

1. Intro to lessons 2-10

1.1. revise input and output

```
var name = prompt('What is your name?');
alert('Hello ' + name + '.\nWelcome to level 2 JavaScript!');
```

1.2. loop revision

Click run, type **20** and click **OK**, then repeat three more times typing the numbers **20**, **20**, and **30**. Type **0** then click **OK**

1.3. while loop revision

```
var x = 0;
while (x <= 8) {
  console.log(x);
  x = x + 2;
}

for (var y = 4; y <= 20; y = y + 4) {
  console.log(y);
}
```

1.4. introduction to string functions

Click run, type your first name and click **OK**, then type your last name and click **OK**.

1.5. introduction to string functions

Click run, type your name, then click **OK**

2. Encoding letters with [[ASCII]] codes

2.1. ASCII codes

```
var answer1 = 32;
var answer2 = 97;
var answer3 = 90;
var answer4 = 48;
```

2.2. ASCII symbols

```
var answer1 = 124;
var answer2 = 42;
var answer3 = 47;
var answer4 = 60;
```

2.3. decode ASCII

```
var answer1 = 'ASCII';
var answer2 = 'Spam';
var answer3 = 'WORM';
var answer4 = 'MaLwArE';
```

2.4. comparing strings

```
var answer1 = true;
var answer2 = true;
var answer3 = false;
var answer4 = false;
```

2.5. ASCII codes

```
var answer1 = 64;
var answer2 = 63;
var answer3 = 95;
var answer4 = 125;
```

3. String [[length]] & [[charAt]]

3.1. string length property

```
var country = prompt('What country do you live in?');
alert(country.length);
```

3.2. use length to check for valid input

```
var country = prompt('What country do you live in?');
if (country.length >= 2 && country.length <= 62) {
  alert('Thanks!');
}
```

3.3. use length to check for valid input

```
var codeName = prompt('Choose your codename');

if (codeName.length < 3) {
  alert('Your codename is too short!');
} else if (codeName.length > 3) {
  alert('Your codename is too long!');
}
```

3.4. use charAt to show each letter in a string

```
var codeName = prompt('Choose your codename');

if (codeName.length < 3) {
  alert('Your codename is too short!');
} else if (codeName.length > 3) {
  alert('Your codename is too long!');
} else {
  alert(codeName.charAt(0));
  alert(codeName.charAt(1));
  alert(codeName.charAt(2));
}
```

3.5. use charAt to check for valid input

```
var codeName = prompt('Choose your codename');

if (codeName.length < 3) {
  alert('Your codename is too short!');
} else if (codeName.length > 3) {
  alert('Your codename is too long!');
} else if (codeName.charAt(0) < 'A' || codeName.charAt(0) > 'Z') {
  alert('The 1st character is invalid!');
} else if (codeName.charAt(1) < 'A' || codeName.charAt(1) > 'Z') {
  alert('The 2nd character is invalid!');
} else if (codeName.charAt(2) < '1' || codeName.charAt(2) > '9') {
  alert('The 3rd character is invalid!');
} else {
  alert('Your codename is valid!');
}
```

4. [[alert]], [[prompt]] & [[string]] review

4.1. review alert

```
alert('Hello');
```

4.2. review prompt and length

```
var name = prompt('Hello, what is your name?');
alert(name.length);
```

4.3.

```
var name = prompt('Hello, what is your name?');
while (name == null) {
    name = prompt('Please tell me your name');
}
alert(name.length);
```

4.4. show a different message depending on **string** length

```
var name = prompt('Hello, what is your name?');
while (name == null) {
    name = prompt('Please tell me your name');
}
if (name.length < 3) {
    alert('Too short!');
} else if (name.length >= 15) {
    alert('Too long!');
} else {
    alert(name.length);
}
```

4.5. check for a valid human name

```
var name = prompt('Hello, what is your name?');
while (name == null) {
    name = prompt('Please tell me your name');
}
var first = name.charAt(0);
var last = name.charAt(name.length - 1);
if (name.length < 3) {
    alert('Too short!');
} else if (name.length >= 15) {
    alert('Too long!');
} else if (first < 'A' || first > 'Z' || last < 'a' || last > 'z') {
    alert('Invalid name');
} else {
    alert(name.length);
}
```

5. Fix broken code

5.1. fix 2 bugs so that the code runs correctly

```
var answer = prompt("Tom's mom has 4 kids: Nickel, Dime, Quarter and who?");
if (answer == 'Tom') {
    alert('Correct! 4 more questions...');
}
```

5.2. find and fix 3 bugs

```
var answer = prompt('What 5 letter word becomes shorter with 2 letters added on?');
if (answer.toLowerCase() == 'short') {
    alert('Correct! 3 questions to go...');
}
```

5.3. find and fix 2 bugs

```
var answer = prompt('A word I know, 6 letters contains. Subtract just 1, & 12 remains. ');
if (answer.toLowerCase() == 'dozens') {
    alert('Correct! 2 to go...');
}
```

5.4. find and fix 2 bugs

```
var answer = prompt('What is the next letter: W I T N ...?');
if (answer == 'L') {
    alert('Correct! Last question coming up...');
}
```

5.5. find and fix 2 bugs

```
var answer = prompt('What object has keys that open no locks, space but no room, and you can enter but not go in?');
if (answer.toLowerCase() == 'keyboard') {
    alert('Correct! Quiz complete!');
}
```

6. Move robots with [[if statements]]

6.1. move the robots to the X

```
robot.forward(2)
if (!robot.onX()) {
    robot.forward(2)
}
if (!robot.onX()) {
    robot.forward(1)
}
if (!robot.onX()) {
    robot.forward(1)
}
```

6.2. move the robots to the X

```
robot.forward(2)
if (!robot.onX()) {
    robot.forward(1)
}
if (!robot.onX()) {
    robot.forward(1)
}
if (!robot.onX()) {
    robot.right(2)
    robot.forward(6)
}
if (!robot.onX()) {
    robot.forward(1)
}
if (!robot.onX()) {
    robot.forward(1)
}
```

6.3. move the robots to the X

```
robot.forward( robot.distanceToX() )
```

6.4. move the robots to the X

```
robot.forward( robot.distanceToX() )
var d = robot.distanceToX()
if (d > 0) {
    robot.right(2)
    robot.forward(d)
}
```

6.5. move the robots to the X

```
var d = robot.distanceToX()
robot.forward(1)
var d2 = robot.distanceToX()
if (d2 > d) {
    robot.right()
    robot.right()
}
robot.forward(d2)
```

7. [[toLowerCase]], [[toUpperCase]], [[substring]]

7.1. toLowerCase and toUpperCase introduction

```
var z = 'Code Avengers';

var a = z.toLowerCase();
var b = z.toUpperCase();
var c = 'Copyright 2012'.toUpperCase();
var d = 'CA'.toLowerCase();

var answerA = 'code avengers';
var answerB = 'CODE AVENGERS';
var answerC = 'COPYRIGHT 2012';
var answerD = 'ca';
```

7.2. substring instruction

```
var x = 'JavaScript';

var a = 'Race'.substring(1);
var b = x.substring(4);
var c = b + ' ' + a;
var d = 'example'.substring(2);

var answerA = 'ace';
var answerB = 'Script';
var answerC = 'Script ace';
var answerD = 'ample';
```

7.3. substring introduction

```
var x = 'Encoding';
var y = 'Nine ninjas';

var a = x.substring(2, 5);
var b = y.substring(3, 10);
var c = a + b;
var d = 'example'.substring(2,4);

var answerA = 'cod';
var answerB = 'e ninja';
var answerC = 'code ninja';
var answerD = 'am';
```

7.4. substring introduction

```
var x = 'Graphic product';
var y = 'Spammer';

var a = x.substring(8, 11);
var b = x.substring(0, 3) + y.substring(3);
var c = a + ' ' + a + b.toLowerCase();

var answerA = 'pro';
var answerB = 'Grammer';
var answerC = 'pro programmer';
```

7.5. charAt introduction

```
var x = 'Rhyming';
var y = 'Waste';
var z = x.charAt(0).toLowerCase();

var a = z + x.substring(4);
var b = x.charAt(3) + y.substring(1) + z;
var c = y.substring(2, y.length - 1) + a + ' ' + b;

var answerA = 'ring';
var answerB = 'master';
var answerC = 'string master';
```

8. String functions practice

8.1. using string functions

Click run, type **Student** and click **OK**. Run the code again but type **Teacher** this time

8.2.

```
var job = prompt('What job would you like to apply for?');

if (job == 'agent') {
    alert('2 jobs available');
} else if (job == 'office') {
    alert('3 jobs available');
} else {
    alert('0 jobs available');
}
```

8.3.

```
var job = prompt('What job would you like to apply for?');
job = job.toLowerCase()

if (job == 'agent') {
    alert('2 jobs available');
} else if (job == 'office') {
    alert('3 jobs available');
} else {
    alert('0 jobs available');
}
```

8.4.

```
var job = prompt('What job would you like to apply for?');
var jobStart = job.substring(0,2).toLowerCase();

if (jobStart == 'ag') {
    alert('2 jobs available');
} else if (jobStart == 'of') {
    alert('3 jobs available');
} else {
    alert('0 jobs available');
}
```

8.5.

```
var job = prompt('What job would you like to apply for?');
var jobStart = job.substring(0,2).toLowerCase();

if (jobStart == 'ag') {
    alert('2 jobs available');
} else if (jobStart == 'of') {
    alert('2 jobs available');
} else if (jobStart == 'en') {
    alert('1 jobs available');
} else if (jobStart == 'cl') {
    alert('1 jobs available');
} else {
    alert('0 jobs available');
}
```

9. Create a multi-choice quiz

9.1. add multi-choice quiz question

```
var guess = prompt('What is phishing?\na. Computer virus\nb. Corrupt website\nc. Junk email\nd. Identity theft scheme');

if (guess == 'd' || guess == 'D') {
    alert('Correct!');
} else {
    alert('Wrong');
}

guess = prompt('What is a firewall?\na. Antivirus program\nb. Internet blocker\nc. Internet filter\nd. Internet logger');

if (guess == 'c' || guess == 'C') {
    alert('Correct!');
} else {
    alert('Wrong');
}
```

9.2. add string quiz question

```

var guess = prompt('What is the 4 letter S word for junk
email or unsolicited bulk email?');

if (guess != null && guess.toLowerCase() == 'spam') {
    alert('Correct!');
} else {
    alert('Wrong');
}

guess = prompt('What 8 letter D word means to retrieve a file
from the internet?');

if (guess != null && guess.toLowerCase() == 'download') {
    alert('Correct!');
} else {
    alert('Wrong');
}

```

9.3. increase a score variable

```

var score = 0;
alert(score);

score = score + 5;
alert(score);

score = score + 5;
alert(score);

score = score + 5;
alert(score);

```

9.4. count the quiz score

```

var score = 0;

var guess = prompt('What is phishing?\na. Computer virus\nb.
Corrupt website\nc. Junk email\nd. Identity theft scheme');

if (guess == 'd' || guess == 'D') {
    alert('Correct!');
    score = score + 1;
} else {
    alert('Wrong');
}

guess = prompt('What is a firewall?\na. Antivirus program\nb.
Internet blocker\nc. Internet filter\nd. Internet logger');

if (guess == 'c' || guess == 'C') {
    alert('Correct!');
    score = score + 1;
} else {
    alert('Wrong');
}

guess = prompt('What is the 4 letter S word for junk email or
unsolicited bulk email?');

if (guess != null && guess.toLowerCase() == 'spam') {
    alert('Correct!');
    score = score + 1;
} else {
    alert('Wrong');
}

guess = prompt('What 8 letter D word means to retrieve a file
from the internet?');

if (guess != null && guess.toLowerCase() == 'download') {
    alert('Correct!');
    score = score + 1;
} else {
    alert('Wrong');
}

alert('You scored: ' + score)

```

9.5. add the final question

```

var score = 0;

var guess = prompt('What is phishing?\na. Computer virus\nb.
Corrupt website\nc. Junk email\nd. Identity theft scheme');

if (guess == 'd' || guess == 'D') {
    alert('Correct!');
    score = score + 1;
} else {
    alert('Wrong');
}

guess = prompt('What is a firewall?\na. Antivirus program\nb.
Internet blocker\nc. Internet filter\nd. Internet logger');

if (guess == 'c' || guess == 'C') {
    alert('Correct!');
    score = score + 1;
} else {
    alert('Wrong');
}

guess = prompt('What is the 4 letter S word for junk email or
unsolicited bulk email?');

if (guess != null && guess.toLowerCase() == 'spam') {
    alert('Correct!');
    score = score + 1;
} else {
    alert('Wrong');
}

guess = prompt('What 8 letter D word means to retrieve a file
from the internet?');

if (guess != null && guess.toLowerCase() == 'download') {
    alert('Correct!');
    score = score + 1;
} else {
    alert('Wrong');
}

guess = prompt('What 6 letter C word stores your settings
when browsing web pages?');

if (guess != null && guess.toLowerCase() == 'cookie') {
    alert('Correct!');
    score = score + 2;
} else {
    alert('Wrong');
}

alert('You scored: ' + score);

```

10. Review strings

10.1. review **string** creation

Review Quiz Questions:

1. ''
2. +
3. ""
4. \n

10.2. review **string functions**

Review Quiz Questions:

1. 'U' < 'Virus'
2. 10
3. a
- 4.

10.3. review **string** more **functions**

Review Quiz Questions:

1. 'FIREWALL'
2. 'wall'
3. 'rewa'
4. 'firewall'

10.4. review using + **strings** and **numbers**

Review Quiz Questions:

1. ab
2. 56
3. 24
4. 33

10.5. final review

Review Quiz Questions:

1. '24'
2. 6
3. CA
4. AVI

11. Intro to lessons 12-20

11.1. introduction to **functions**

```
var response = prompt('What do you want to learn next?');

if(response == 'functions') {
  alert('Let\'s go!');
}
```

11.2. **call** a function and **comment out**

```
//Simple function
function testFunction() {
  alert('This message is inside the function');
}

//testFunction();
```

11.3. introduction to **vector graphics** with [Paper.js](#)

```
var SIZE = view.size.height;
var GRAVITY = -2;
var BOUNCE = -0.9;
var RADIUS = 40;
var COLOR1 = 'yellow';
var COLOR2 = 'black';

//Function to create a ball
function Ball(point) {
  this.velocity = 0;
  this.item = new Path.Circle(point, RADIUS)
  this.item.fillColor = new GradientColor(
    new Gradient([COLOR1, COLOR2], 'radial'),
    point, point + RADIUS, point + RADIUS / 8);
}

//Function to move each ball each frame
function move(ball) {
  ball.velocity += GRAVITY;

  var y = ball.item.position.y + ball.velocity;

  if (y < RADIUS || y > SIZE - RADIUS) {
    ball.velocity *= BOUNCE;
  } else {
    ball.item.position.y = y;
  }
}

//Create a list of balls
var balls = [];

//This function is called automatically by Paper.js when the
mouse is clicked
function onMouseUp(event) {
  var ball = new Ball(event.point);
  balls.push(ball);
}

//This function is called automatically by Paper.js 60 times
per second
function onFrame() {
  for (var i = 0, l = balls.length; i < l; i++) {
    move(balls[i]);
  }
}
```

11.4. modify another Paper.js example

```
var mousePoint = view.center;
var amount = 10;
var colors = ['black', 'red', 'black', 'gold'];
var children = project.activeLayer.children;

for (var i = 0; i < amount; i++) {
  var rect = new Rectangle([0, 0], [10, 10]);
  rect.center = mousePoint;
  var path = new Path.RoundRectangle(rect, 2);
  path.fillColor = colors[i % 4];
  var scale = (1 - i / amount) * 20;
  path.scale(scale);
}

function onMouseMove(event) {
  mousePoint = event.point;
}

function onFrame(event) {
  for (var i = 0, l = children.length; i < l; i++) {
    var item = children[i];
    var delta = (mousePoint - item.position) / (i + 5);
    item.rotate(Math.sin((event.count + i) / 10) * 5);
    if (delta.length > 0.1)
      item.translate(delta);
  }
}
```

11.5. move Paper.js objects

```
function drawBrick(point, size, count) {
  var path = new Path.Rectangle(point, size);
  path.strokeColor = 'orange';
  path.fillColor = 'yellow';
  path.strokeWidth = 2;
}

var size = new Size(100, 50);

drawBrick(new Point(150, 250), size);
drawBrick(new Point(250, 250), size);
drawBrick(new Point(200, 200), size);
```

12. Functions introduction

12.1. review functions from level 1

```
var itemCount1 = prompt('Enter item count 1:');
var itemCount2 = prompt('Enter item count 2:');
var itemCount3 = prompt('Enter item count 3:');

var itemTotal = Number(itemCount1) + Number(itemCount2) +
Number(itemCount3);

alert('Total items:' + itemTotal);
```

12.2. use a simple function

```
//Calculates the total items
function getItemTotal() {

  var itemCount1 = prompt('Enter item count 1:');
  var itemCount2 = prompt('Enter item count 2:');
  var itemCount3 = prompt('Enter item count 3:');

  var itemTotal = Number(itemCount1) + Number(itemCount2) +
Number(itemCount3);

  alert('Total items:' + itemTotal);
}

getItemTotal();
getItemTotal();
```

12.3. use a function with a parameter

```
//Calculates the total of n item counts
function getItemTotal2(n) {
  var total = 0;

  for (var i = 1; i <= n; i++) {
    var count = prompt('Enter item count ' + i);
    total += Number(count);
  }

  alert('Total items:' + total);
}

//Calculates the total of 3 item counts
function getItemTotal() {
  var itemCount1 = prompt('Enter item count 1:');
  var itemCount2 = prompt('Enter item count 2:');
  var itemCount3 = prompt('Enter item count 3:');

  var itemTotal = Number(itemCount1) + Number(itemCount2) +
Number(itemCount3);

  alert('Total items:' + itemTotal);
}

getItemTotal2(2);
getItemTotal2(4);
```

12.4. use a function with 2 parameters

```
//Calculate the cost of equipment for each Lab
function getTotalCost(labCount, itemCost) {
  var totalCost = 0;

  for (var i = 1; i <= labCount; i++) {
    var count = prompt('Item count for lab ' + i);
    totalCost += Number(count);

    var cost = count * itemCost;
    console.log('Lab ' + i + ' cost: $' + cost);
  }

  var totalCost = totalCost * itemCost;
  console.log('Total cost: $' + totalCost);
  console.log('Total items: ' + totalCost);
}

getTotalCost(5, 875);
```

12.5. use a function with a return value

```
//Calculate the cost of equipment for each Lab
function getTotalCost(labCount, itemCost) {
  var totalCost = 0;

  for (var i = 1; i <= labCount; i++) {
    var count = prompt('Item count for lab ' + i);
    totalCost += Number(count);

    var cost = count * itemCost;
    console.log('Lab ' + i + ' cost: $' + cost);
  }

  return totalCost * itemCost;
}

var computerCost = getTotalCost(5, 875);
console.log('Computer cost: $' + computerCost);

var deskCost = getTotalCost(3, 89);
console.log('Desk cost: $' + deskCost);

var totalCost = deskCost + computerCost;
console.log('Total cost: $' + totalCost);
```

13. Learn to understand functions

13.1. determine return values of functions with 1 parameter

13.2. determine return values of functions with 2 parameters

13.3. determine return values of functions with if statements

13.4. determine return values of boolean functions with 1 parameter

13.5. determine return values of boolean functions with 1 parameter

14. Intro to drawing

14.1. create a Path object to draw a straight line

```
var p = new Path();
p.strokeColor = 'red';
p.add(new Point(50, 50));
p.add(new Point(450, 50));
```

14.2. use a Path object to draw a square

```
var p = new Path();
p.strokeColor = 'green';
p.add(new Point(50, 50));
p.add(new Point(450, 50));
p.add(new Point(450, 450));
p.add(new Point(50, 450));
p.closed = true
```

14.3. use the strokeWidth property to increase the width of a line

```

var p = new Path();
p.strokeColor = 'green';
p.add(new Point(50, 50));
p.add(new Point(450, 50));
p.add(new Point(450, 450));
p.add(new Point(50, 450));
p.closed = true
p.strokeWidth = 5;

p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(250, 50));
p.add(new Point(250, 450));

p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(50, 250));
p.add(new Point(450, 250));

```

14.4. create circle objects

```

var p = new Path();
p.strokeColor = 'green';
p.add(new Point(50, 50));
p.add(new Point(450, 50));
p.add(new Point(450, 450));
p.add(new Point(50, 450));
p.closed = true
p.strokeWidth = 5;

//Draw roads
p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(250, 50));
p.add(new Point(250, 450));

p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(50, 250));
p.add(new Point(450, 250));

//Draw roundabout
var center = new Point(250, 250);
var roundabout = new Path.Circle(center, 10);
roundabout.strokeWidth = 10;
roundabout.strokeColor = 'black';

//Draw main office
var center = new Point(150, 150);
var mainOffice = new Path.Circle(center, 60);
mainOffice.strokeWidth = 3;
mainOffice.strokeColor = 'red';

```

14.5. create regular polygon objects

```

var p = new Path();
p.strokeColor = 'green';
p.add(new Point(50, 50));
p.add(new Point(450, 50));
p.add(new Point(450, 450));
p.add(new Point(50, 450));
p.closed = true
p.strokeWidth = 5;

//Draw roads
p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(250, 50));
p.add(new Point(250, 450));

p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(50, 250));
p.add(new Point(450, 250));

//Draw roundabout
var center = new Point(250, 250);
var roundabout = new Path.Circle(center, 10);
roundabout.strokeWidth = 10;
roundabout.strokeColor = 'black';

//Draw the main office
center = new Point(150, 150);
var mainOffice = new Path.RegularPolygon(center, 5, 60);
mainOffice.strokeWidth = 3;
mainOffice.strokeColor = 'red';

//Draw the plaza
var plaza = new Path.RegularPolygon(center, 5, 20);
plaza.strokeWidth = 3;
plaza.strokeColor = 'green';

```

15. Fix broken functions

15.1. fix bugs in function code

```

function getTotalCost() {
  var totalCost = 0;
  var cost = prompt('Enter the first cost: ');

  while (cost > 0) {
    totalCost += Number(cost);
    cost = prompt('Enter the next cost: ');
  }

  return totalCost;
}

var budget = prompt('Enter the budget: ');
var totalCosts = getTotalCost();

```

15.2. fix **syntax errors**

```
function getTotalCost() {
  var totalCost = 0;
  var cost = prompt('Enter the first cost: ');

  while (cost > 0) {
    totalCost += Number(cost);
    cost = prompt('Enter the next cost: ');
  }

  return totalCost;
}

function displayReport(budget, costs) {
  var balance = budget - costs;
  console.log('Budget: $' + budget);
  console.log('Costs: $' + costs);
  console.log('Balance: $' + balance);
}

var budget = prompt('Enter the budget: ');
var totalCost = getTotalCost();
displayReport(budget, totalCost);
```

15.3. fix **logic errors**

```
//Returns the minimum value of a and b
function min(a, b) {
  if (a > b) {
    return b;
  }

  return a;
}

//Returns the average of 3 values
function average(a, b, c) {
  return (a + b + c) / 3;
}

console.log(min(5, 10));
console.log(min(25, 20));
console.log(min(15, 15));

console.log(average(15, 25, 35));
```

15.4. fix **syntax errors** in function code

```
//Returns the minimum value of a and b
function min(a, b) {
  if (a > b) {
    return b;
  }

  return a;
}

//Returns the average of 3 values
function average(a, b, c) {
  return (a + b + c) / 3;
}

console.log(min(5, 10));
console.log(min(25, 20));
console.log(min(15, 15));

console.log(average(15, 25, 35));
```

15.5. fix **syntax errors** in function calls

```
//Returns the minimum value of a and b
function min(a, b) {
  if (Number(a) > Number(b)) {
    return b;
  }

  return a;
}

//Returns the average of 3 values
function average(a, b, c) {
  return (Number(a) + Number(b) + Number(c)) / 3;
}

var companyA = prompt('Enter quote for company A');
var companyB = prompt('Enter quote for company B');
var companyC = prompt('Enter quote for company C');

var averageQuote = average(companyA, companyB, companyC);

var minAB = min(companyA, companyB);
var minQuote = min(minAB, companyC);

console.log('Min quote: ' + minQuote);
console.log('Average quote: $' + averageQuote.toFixed(2));
```

16. Variable scope and parameters

16.1. determine the value of variables changed by a function

16.2. determine the value of variables when a function runs twice

16.3. variables and parameters with the same name

16.4. calling functions with variables

16.5. function variables

17. Drawing rectangles

17.1. create rectangle objects

```

var p = new Path();
p.strokeColor = 'green';
p.add(new Point(50, 50));
p.add(new Point(450, 50));
p.add(new Point(450, 450));
p.add(new Point(50, 450));
p.closed = true
p.strokeWidth = 5;

//Draw roads
p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(250, 50));
p.add(new Point(250, 450));

p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(50, 250));
p.add(new Point(450, 250));

//Draw roundabout
var center = new Point(250, 250);
p = new Path.Circle(center, 10);
p.strokeWidth = 10;
p.strokeColor = 'black';

//Draw the main office
center = new Point(150, 150);
var mainOffice = new Path.RegularPolygon(center, 5, 60);
mainOffice.strokeWidth = 2;
mainOffice.strokeColor = 'red';

var plaza = new Path.RegularPolygon(center, 5, 20);
plaza.strokeWidth = 2;
plaza.strokeColor = 'green';

//Draw the car park
var tl = new Point(60, 260);
var br = new Point(240, 350);
var carPark = new Path.Rectangle(tl, br);
carPark.strokeWidth = 2;
carPark.strokeColor = 'black';

//Draw the research lab
tl = new Point(80, 370);
br = new Point(120, 430);
var lab = new Path.Rectangle(tl, br);
lab.strokeWidth = 2;
lab.strokeColor = 'brown';

```

17.2. draw several rectangles with a loop

```

var p = new Path();
p.strokeColor = 'green';
p.add(new Point(50, 50));
p.add(new Point(450, 50));
p.add(new Point(450, 450));
p.add(new Point(50, 450));
p.closed = true
p.strokeWidth = 5;

//Draw roads
p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(250, 50));
p.add(new Point(250, 450));

p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(50, 250));
p.add(new Point(450, 250));

//Draw roundabout
var center = new Point(250, 250);
p = new Path.Circle(center, 10);
p.strokeWidth = 10;
p.strokeColor = 'black';

//Draw the main office
center = new Point(150, 150);
var mainOffice = new Path.RegularPolygon(center, 5, 60);
mainOffice.strokeWidth = 2;
mainOffice.strokeColor = 'red';

var plaza = new Path.RegularPolygon(center, 5, 20);
plaza.strokeWidth = 2;
plaza.strokeColor = 'green';

//Draw the car park
var tl = new Point(60, 260);
var br = new Point(240, 350);
var carPark = new Path.Rectangle(tl, br);
carPark.strokeWidth = 2;
carPark.strokeColor = 'black';

//Draw the 3 research labs
for (var i = 0; i < 3; i++) {
  //Draw a Lab
  var tl = new Point(80 + i * 50, 370);
  var br = new Point(120 + i * 50, 430);
  var lab = new Path.Rectangle(tl, br);
  lab.strokeWidth = 2;
  lab.strokeColor = 'brown';
}

```

17.3. draw filled shapes

```

var p = new Path();
p.strokeColor = 'green';
p.add(new Point(50, 50));
p.add(new Point(450, 50));
p.add(new Point(450, 450));
p.add(new Point(50, 450));
p.closed = true
p.strokeWidth = 5;

//Draw roads
p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(250, 50));
p.add(new Point(250, 450));

p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(50, 250));
p.add(new Point(450, 250));

//Draw roundabout
var center = new Point(250, 250);
p = new Path.Circle(center, 10);
p.strokeWidth = 10;
p.strokeColor = 'black';

//Draw the main office
center = new Point(150, 150);
var mainOffice = new Path.RegularPolygon(center, 5, 60);
mainOffice.strokeWidth = 2;
mainOffice.strokeColor = 'red';

var plaza = new Path.RegularPolygon(center, 5, 20);
plaza.strokeWidth = 2;
plaza.strokeColor = 'green';

//Draw the car park
var tl = new Point(60, 260);
var br = new Point(240, 350);
var carPark = new Path.Rectangle(tl, br);
carPark.strokeWidth = 2;
carPark.strokeColor = 'black';

//Draw the 3 research labs
for (var i = 0; i < 3; i++) {
  //Draw a Lab
  var tl = new Point(80 + i * 50, 370);
  var br = new Point(120 + i * 50, 430);
  var lab = new Path.Rectangle(tl, br);
  lab.strokeWidth = 2;
  lab.strokeColor = 'brown';
}

//Draw the Lake
center = new Point(350, 350);
var lake = new Path.Circle(center, 30);
lake.strokeWidth = 2;
lake.strokeColor = 'blue';
lake.fillColor = 'lightBlue';

```

17.4. practice filling shapes

```

var p = new Path();
p.strokeColor = 'green';
p.add(new Point(50, 50));
p.add(new Point(450, 50));
p.add(new Point(450, 450));
p.add(new Point(50, 450));
p.closed = true
p.strokeWidth = 5;

//Draw roads
p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(250, 50));
p.add(new Point(250, 450));

p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(50, 250));
p.add(new Point(450, 250));

//Draw roundabout
var center = new Point(250, 250);
p = new Path.Circle(center, 10);
p.strokeWidth = 10;
p.strokeColor = 'black';

//Draw the main office
center = new Point(150, 150);
var mainOffice = new Path.RegularPolygon(center, 5, 60);
mainOffice.strokeWidth = 2;
mainOffice.strokeColor = 'red';
mainOffice.fillColor = 'pink';

var plaza = new Path.RegularPolygon(center, 5, 20);
plaza.strokeWidth = 2;
plaza.strokeColor = 'green';
plaza.fillColor = 'lightGreen';

//Draw the car park
var tl = new Point(60, 260);
var br = new Point(240, 350);
var carPark = new Path.Rectangle(tl, br);
carPark.strokeWidth = 2;
carPark.strokeColor = 'black';
carPark.fillColor = 'darkGray';

//Draw the 3 research Labs
for (var i = 0; i < 3; i++) {
  //Draw a Lab
  var tl = new Point(80 + i * 50, 370);
  var br = new Point(120 + i * 50, 430);
  var lab = new Path.Rectangle(tl, br);
  lab.strokeWidth = 2;
  lab.strokeColor = 'brown';
  lab.fillColor = 'orange';
}

//Draw the Lake
center = new Point(350, 350);
var lake = new Path.Circle(center, 30);
lake.strokeWidth = 2;
lake.strokeColor = 'blue';
lake.fillColor = 'lightBlue';

```

17.5. practice drawing shapes

```

var p = new Path();
p.strokeColor = 'green';
p.add(new Point(50, 50));
p.add(new Point(450, 50));
p.add(new Point(450, 450));
p.add(new Point(50, 450));
p.closed = true
p.strokeWidth = 5;

//Draw roads
p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(250, 50));
p.add(new Point(250, 450));

p = new Path();
p.strokeWidth = 10;
p.strokeColor = 'black';
p.add(new Point(50, 250));
p.add(new Point(450, 250));

//Draw roundabout
var center = new Point(250, 250);
p = new Path.Circle(center, 10);
p.strokeWidth = 10;
p.strokeColor = 'black';

//Draw the main office
center = new Point(150, 150);
var mainOffice = new Path.RegularPolygon(center, 5, 60);
mainOffice.strokeWidth = 2;
mainOffice.strokeColor = 'red';
mainOffice.fillColor = 'pink';

var plaza = new Path.RegularPolygon(center, 5, 20);
plaza.strokeWidth = 2;
plaza.strokeColor = 'green';
plaza.fillColor = 'lightGreen';

//Draw the car park
var tl = new Point(60, 260);
var br = new Point(240, 350);
var carPark = new Path.Rectangle(tl, br);
carPark.strokeWidth = 2;
carPark.strokeColor = 'black';
carPark.fillColor = 'darkGray';

//Draw the 3 research Labs
for (var i = 0; i < 3; i++) {
  //Draw a Lab
  var tl = new Point(80 + i * 50, 370);
  var br = new Point(120 + i * 50, 430);
  var lab = new Path.Rectangle(tl, br);
  lab.strokeWidth = 2;
  lab.strokeColor = 'brown';
  lab.fillColor = 'orange';
}

//Draw the Lake
center = new Point(350, 350);
var lake = new Path.Circle(center, 30);
lake.strokeWidth = 2;
lake.strokeColor = 'blue';
lake.fillColor = 'lightBlue';

//Draw the sports fields
tl = new Point(350, 60);
br = new Point(440, 240);
var fields = new Path.Rectangle(tl, br);
fields.strokeWidth = 2;
fields.strokeColor = 'green';
fields.fillColor = 'lightGreen';

//Draw the aquatic center
center = new Point(300, 100);
var pools = new Path.RegularPolygon(center, 4, 55);
pools.strokeWidth = 2;
pools.strokeColor = 'blue';
pools.fillColor = 'lightBlue';

//Draw the gymnasium
center = new Point(300, 195);
var gym = new Path.RegularPolygon(center, 4, 55);

```

```

gym.strokeWidth = 2;
gym.strokeColor = 'gold';
gym.fillColor = 'yellow';

```

18. Functions practice

18.1. complete the function

```

//Calculate the area of a rectangle
function calculateArea(length, width) {
  return length * width;
}

var ceilingArea1 = calculateArea(8, 5);
var ceilingArea2 = calculateArea(12, 6);
var ceilingArea3 = calculateArea(8, 7);

var totalCeilingArea = ceilingArea1 + ceilingArea2 +
ceilingArea3;
var paintCost = totalCeilingArea * 12.5;

console.log('Ceiling area: ' + totalCeilingArea);
console.log('Paint cost: $' + paintCost.toFixed(2));

```

18.2. complete the function

```

//Calculate the area of a rectangle
function calculateArea(length, width) {
  return length * width;
}

//Calculate the total area of the 4 walls in a room
function calculateWallArea(length, width, height) {
  return calculateArea(length, height) * 2 +
calculateArea(width, height) * 2;
}

//Calculate painting costs for walls and ceiling
function calculateCostToPaint(length, width, height) {
  return calculateWallArea(length, width, height) * 13.5 +
calculateArea(length, width) * 12.5;
}

var costLab1 = calculateCostToPaint(8, 5, 3);
var costLab2 = calculateCostToPaint(12, 6, 3);
var costLab3 = calculateCostToPaint(8, 7, 3.5);

var totalCost = costLab1 + costLab2 + costLab3;
console.log('Total cost: $' + totalCost.toFixed(2));

```

18.3. update the function

```

//Calculate the area of a rectangle
function calculateArea(length, width) {
  return length * width;
}

//Calculate the total area of the 4 walls in a room
function calculateWallArea(length, width, height) {
  return calculateArea(length, height) * 2 +
calculateArea(width, height) * 2;
}

//Calculate painting costs for walls and ceiling
function calculateCostToPaint(length, width, height) {
  return calculateWallArea(length, width, height) * 11 +
calculateArea(length, width) * 11;
}

var costLab1 = calculateCostToPaint(8, 5, 3);
var costLab2 = calculateCostToPaint(12, 6, 3);
var costLab3 = calculateCostToPaint(8, 7, 3.5);
var costLab4 = calculateCostToPaint(6, 5, 3.5);

var totalCost = costLab1 + costLab2 + costLab3 + costLab4;
console.log('Total cost: $' + totalCost.toFixed(2));

```

18.4. complete the function

```
//Calculate the area of a rectangle
function calculateArea(length, width) {
  return length * width;
}

//Calculate the cost of installing carpet
function calculateCarpetCost(fixed, variable, area) {
  return fixed + variable * area;
}

var floorArea1 = calculateArea(8, 5);
var floorArea2 = calculateArea(12, 6);
var floorArea3 = calculateArea(8, 7);
var floorArea4 = calculateArea(6, 5);

var totalFloorArea = floorArea1 + floorArea2 + floorArea3 +
floorArea4;

var costA = calculateCarpetCost(0, 40, totalFloorArea);
var costB = calculateCarpetCost(45, 38, totalFloorArea);
var costC = calculateCarpetCost(500, 35, totalFloorArea);

console.log('Carpet cost A: $' + costA);
console.log('Carpet cost B: $' + costB);
console.log('Carpet cost C: $' + costC);
```

18.5. create a function

```
//Calculate the area of a rectangle
function calculateArea(length, width) {
  return length * width;
}

//Calculate the cost of installing carpet
function calculateCarpetCost(fixed, variable, area) {
  return fixed + variable * area;
}

var floorArea1 = calculateArea(8, 5);
var floorArea2 = calculateArea(12, 6);
var floorArea3 = calculateArea(8, 7);
var floorArea4 = calculateArea(6, 5);

var totalFloorArea = floorArea1 + floorArea2 + floorArea3 +
floorArea4;

var costA = calculateCarpetCost(0, 40, totalFloorArea);
var costB = calculateCarpetCost(45, 38, totalFloorArea);
var costC = calculateCarpetCost(200, 35, totalFloorArea);

console.log('Carpet cost A: $' + costA);
console.log('Carpet cost B: $' + costB);
console.log('Carpet cost C: $' + costC);

//Calculate the lowest value of a, b and c
function min(a, b, c) {
  if(a < b && a < c) {
    return a;
  } else if (b < c) {
    return b;
  }
  return c;
}

var minCost = min(costA, costB, costC);
console.log('Lowest cost: $' + minCost);
```

19. Functions practice

19.1. complete calculateArea

```
//Calculate the area of a rectangle
function calculateArea(length, width) {
  return length * width;
}

var bathroom1 = calculateArea(5, 3);
var bathroom2 = calculateArea(6.5, 3.5);
var bathroom3 = calculateArea(9, 4);
var bathroom4 = calculateArea(11, 4);

console.log('Bathroom Tiling' +
'\nBathroom 1: ' + bathroom1 +
'\nBathroom 2: ' + bathroom2 +
'\nBathroom 3: ' + bathroom3 +
'\nBathroom 4: ' + bathroom4);
```

19.2. create calculateTileCount

```
//Calculate the area of a rectangle
function calculateArea(length, width) {
  return length * width;
}

function calculateTileCount(tileWidth, length, width) {
  var tileArea = tileWidth * tileWidth;
  var tileCount = calculateArea(length, width) / tileArea;
  return tileCount;
}

var bathroom1 = calculateTileCount(0.3, 5, 3);
var bathroom2 = calculateTileCount(0.3, 6.5, 3.5);
var bathroom3 = calculateTileCount(0.45, 9, 4);
var bathroom4 = calculateTileCount(0.45, 11, 4);

console.log('Bathroom Tiling' +
'\nBathroom 1: ' + bathroom1.toFixed(0) +
'\nBathroom 2: ' + bathroom2.toFixed(0) +
'\nBathroom 3: ' + bathroom3.toFixed(0) +
'\nBathroom 4: ' + bathroom4.toFixed(0));
```

19.3. create calculateTileCost

```
//Calculate the area of a rectangle
function calculateArea(length, width) {
  return length * width;
}

function calculateTileCount(tileWidth, length, width) {
  var tileArea = tileWidth * tileWidth;
  var tileCount = calculateArea(length, width) / tileArea;
  return tileCount;
}

function calculateTileCost(costPerTile, tileWidth, length,
width) {
  var totalTileCost = calculateTileCount(tileWidth, length,
width) * costPerTile;
  return totalTileCost;
}

var bathroom1 = calculateTileCost(1.5, 0.3, 5, 3);
var bathroom2 = calculateTileCost(1.5, 0.3, 6.5, 3.5);
var bathroom3 = calculateTileCost(3.5, 0.45, 9, 4);
var bathroom4 = calculateTileCost(3.5, 0.45, 11, 4);

console.log('Bathroom Tiling' +
'\nBathroom 1: $' + bathroom1.toFixed(2) +
'\nBathroom 2: $' + bathroom2.toFixed(2) +
'\nBathroom 3: $' + bathroom3.toFixed(2) +
'\nBathroom 4: $' + bathroom4.toFixed(2));
```

19.4. create calculateTilingCost

```
//Calculate the area of a rectangle
function calculateArea(length, width) {
  return length * width;
}

function calculateTileCount(tileWidth, length, width) {
  var tileArea = tileWidth * tileWidth;
  var tileCount = calculateArea(length, width) / tileArea;
  return tileCount;
}

function calculateTileCost(costPerTile, tileWidth, length, width) {
  var totalTileCost = calculateTileCount(tileWidth, length, width) * costPerTile;
  return totalTileCost;
}

function calculateTilingCost(costPerTile, tileWidth, length, width) {
  var totalCost = calculateTileCost(costPerTile, tileWidth, length, width) + calculateArea(length, width) * 18 + 60;
  return totalCost;
}

var bathroom1 = calculateTilingCost(1.5, 0.3, 5, 3);
var bathroom2 = calculateTilingCost(1.5, 0.3, 6.5, 3.5);
var bathroom3 = calculateTilingCost(3.5, 0.45, 9, 4);
var bathroom4 = calculateTilingCost(3.5, 0.45, 11, 4);

console.log('Bathroom Tiling' +
  '\nBathroom 1: $' + bathroom1.toFixed(2) +
  '\nBathroom 2: $' + bathroom2.toFixed(2) +
  '\nBathroom 3: $' + bathroom3.toFixed(2) +
  '\nBathroom 4: $' + bathroom4.toFixed(2));
```

19.5. reuse the functions

```
//Calculate the area of a rectangle
function calculateArea(length, width) {
  return length * width;
}

function calculateTileCount(tileWidth, length, width) {
  var tileArea = tileWidth * tileWidth;
  var tileCount = calculateArea(length, width) / tileArea;
  return tileCount;
}

function calculateTileCost(costPerTile, tileWidth, length, width) {
  var totalTileCost = calculateTileCount(tileWidth, length, width) * costPerTile;
  return totalTileCost;
}

function calculateTilingCost(costPerTile, tileWidth, length, width) {
  var totalCost = calculateTileCost(costPerTile, tileWidth, length, width) + calculateArea(length, width) * 18 + 60;
  return totalCost;
}

var bathroom1 = calculateTilingCost(1.5, 0.3, 5, 3);
var bathroom2 = calculateTilingCost(1.5, 0.3, 6.5, 3.5);
var bathroom3 = calculateTilingCost(3.5, 0.45, 9, 4);
var bathroom4 = calculateTilingCost(3.5, 0.45, 11, 4);

var bathroom1b = calculateTilingCost(2, 0.3, 5, 3);
var bathroom2b = calculateTilingCost(2, 0.3, 6.5, 3.5);

console.log('Bathroom Tiling' +
  '\nBathroom 1: $' + bathroom1.toFixed(2) +
  '\nBathroom 2: $' + bathroom2.toFixed(2) +
  '\nBathroom 3: $' + bathroom3.toFixed(2) +
  '\nBathroom 4: $' + bathroom4.toFixed(2));

var totalCostA = bathroom1 + bathroom2 + bathroom3 + bathroom4;
console.log('Total cost A: $' + totalCostA);

var totalCostB = bathroom1b + bathroom2b + bathroom3 + bathroom4;

console.log('Total cost B: $' + totalCostB);
```

20. Review lessons 11-19

20.1. review functions that do calculations

Review Quiz Questions:

1. mean
2. max
3. min
4. sum

20.2. review functions that return boolean values

Review Quiz Questions:

1. true false
2. false true
3. false false
4. true true

20.3. review functions that check a character variable

Review Quiz Questions:

1. isUpperCase
2. isVowel
3. isConsonant
4. isLetter

20.4. review function return values

Review Quiz Questions:

1. 10
2. 24
3. 10
4. 21

20.5. review variable and parameters

Review Quiz Questions:

1. 130
2. 120
3. 5
4. 20

21. Intro to lessons 22-30

21.1. introduction to Math.pow

```
console.log(2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2);
console.log( Math.pow(2, 10) );
```

21.2. introduction to Math.abs and Math.sqrt

```
console.log( Math.sqrt(256) );
console.log( Math.abs(-256) );
console.log( Math.abs(256) );
```

21.3. introduction to Math.PI

```
console.log( Math.pow(Math.PI, 2) );
console.log( 2.0 * Math.PI * Math.pow(0.8, 2) );
```

21.4. introduction to Math.round

```
console.log( Math.round(Math.pow(Math.PI, 2)) );
console.log( Math.round(2.0 * Math.PI * Math.pow(0.8, 2)) );
```

21.5. introduction to Math.random

```
console.log( Math.random() );
console.log( Math.random() );
console.log( Math.random() );
console.log( Math.random() );
console.log( Math.random() );
```

22. Use Math object functions

22.1. use Math functions

```
var tankVolume = Math.pow(25, 3);
var logoArea = Math.PI * Math.pow(11, 2);
var landWidth = Math.sqrt(375000);

console.log( 'Tank volume: ' + tankVolume );
console.log( 'Logo area: ' + logoArea );
console.log( 'Land width: ' + landWidth );
```

22.2. use Math functions

```
var tankHeatingCost = Math.pow(25, 3) / 15;
var logoPaintCost = Math.PI * Math.pow(11, 2) * 2;
var landPerimeter = Math.sqrt(375000) * 4;

console.log( 'Tank heating cost: $' + tankHeatingCost );
console.log( 'Logo paint cost: $' + logoPaintCost );
console.log( 'Land perimeter: ' + landPerimeter );
```

22.3. use Math functions

```
var tankHeatingCost = Math.pow(25, 3) / 15;
var logoPaintCost = Math.PI * Math.pow(11, 2) * 2;
var landPerimeter = Math.sqrt(375000) * 4;

console.log( 'Tank heating cost: $' +
Math.floor(tankHeatingCost) );
console.log( 'Logo paint cost: $' + Math.ceil(logoPaintCost)
);
console.log( 'Land perimeter: ' + Math.round(landPerimeter)
);
```

22.4. round numbers with toFixed

```
var tankHeatingCost = Math.pow(25, 3) / 15;
var logoPaintCost = Math.PI * Math.pow(11, 2) * 2;
var landPerimeter = Math.sqrt(375000) * 4;

console.log( 'Tank heating cost: $' +
tankHeatingCost.toFixed(2) );
console.log( 'Logo paint cost: $' + logoPaintCost.toFixed(2)
);
console.log( 'Land perimeter: ' + landPerimeter.toFixed(0) );
```

22.5. use toPrecision

```
var tankHeatingCost = Math.pow(25, 3) / 15;
var logoPaintCost = Math.PI * Math.pow(11, 2) * 2;
var landPerimeter = Math.sqrt(375000) * 4;

console.log( 'Tank heating cost: $' +
Math.round(tankHeatingCost.toPrecision(2)) );
console.log( 'Logo paint cost: $' +
Math.round(logoPaintCost.toPrecision(2)) );
console.log( 'Land perimeter: ' +
Math.round(landPerimeter.toPrecision(3)) );
```

23. Generate random numbers

23.1. random number from 0-1

```
console.log( Math.random() );
console.log( Math.random() );
console.log( Math.random() );
```

23.2. random integers

```
console.log( Math.floor(38 * Math.random()) );
console.log( Math.floor(71 * Math.random()) );
console.log( Math.floor(360 * Math.random()) );
```

23.3. random integers in a range

```
console.log( Math.floor(1 + 6 * Math.random()) );
console.log( Math.floor(30 + 41 * Math.random()) );
console.log( Math.floor(-180 + 360 * Math.random()) );
```

23.4. random multiples

```
console.log( Math.floor(1 + 6 * Math.random()) +
Math.floor(1 + 6 * Math.random()) );
console.log(5 * Math.floor(21 * Math.random()) );
console.log(10 * Math.floor(37 * Math.random()) );
```

23.5. random workout

```
//Random workout generator
var runTime = 10 + Math.floor(21 * Math.random());
var situpCount = 50 + 10 * Math.floor(6 * Math.random());
var pushupCount = 200 - situpCount;

alert('Run: ' + runTime + 'mins' +
'\nSitups: ' + situpCount +
'\nPushups: ' + pushupCount);
```

24. Draw the Pyramids of Giza

24.1. move 3 blocks

```
//Draw the bottom left brick
var tl = new Point(10, 460);
var br = new Point(40, 490);
var brick1 = new Path.Rectangle(tl, br);
brick1.strokeWidth = 2;
brick1.strokeColor = 'gold';
brick1.fillColor = 'yellow';

//Draw the bottom right brick
tl = new Point(40, 460);
br = new Point(70, 490);
var brick2 = new Path.Rectangle(tl, br);
brick2.strokeWidth = 2;
brick2.strokeColor = 'gold';
brick2.fillColor = 'yellow';

//Draw the top brick
tl = new Point(25, 430);
br = new Point(55, 460);
var brick3 = new Path.Rectangle(tl, br);
brick3.strokeWidth = 2;
brick3.strokeColor = 'gold';
brick3.fillColor = 'yellow';
```

24.2. use a function to draw blocks

```
//Draw 1 block at location x, y
function drawBlock(x, y) {
  var tl = new Point(x, y);
  var br = new Point(x + 30, y + 30);
  var brick = new Path.Rectangle(tl, br);
  brick.strokeWidth = 2;
  brick.strokeColor = 'gold';
  brick.fillColor = 'yellow';
}

//Draw the 3 bricks
drawBlock(10, 460); //brick 1
drawBlock(40, 460); //brick 2
drawBlock(25, 430); //brick 3
```

24.3. use a loop to draw a row of blocks

```
//Draw 1 block at location x, y
function drawBlock(x, y) {
  var tl = new Point(x, y);
  var br = new Point(x + 30, y + 30);
  var brick = new Path.Rectangle(tl, br);
  brick.strokeWidth = 2;
  brick.strokeColor = 'gold';
  brick.fillColor = 'yellow';
}

//Draw the row of bricks
for(var i = 0; i < 16; i++) {
  drawBlock(10 + i * 30, 460);
}
```

24.4. draw several rows of blocks

```
//Draw 1 block at location x, y
function drawBlock(x, y) {
  var tl = new Point(x, y);
  var br = new Point(x + 30, y + 30);
  var brick = new Path.Rectangle(tl, br);
  brick.strokeWidth = 2;
  brick.strokeColor = 'gold';
  brick.fillColor = 'yellow';
}

//Draw the rows of bricks
for(var i = 0; i < 16; i++) {
  drawBlock(10 + i * 30, 460);
}

for(var i = 0; i < 15; i++) {
  drawBlock(25 + i * 30, 430);
}

for(var i = 0; i < 14; i++) {
  drawBlock(40 + i * 30, 400);
}

for(var i = 0; i < 13; i++) {
  drawBlock(55 + i * 30, 370);
}
```

24.5. practice drawing shapes

```
var BLOCK_SIZE = 10;

//Draw 1 block at location x, y
function drawBlock(x, y) {
  var tl = new Point(x, y);
  var br = new Point(x + BLOCK_SIZE, y + BLOCK_SIZE);
  var brick = new Path.Rectangle(tl, br);
  brick.strokeWidth = 2;
  brick.strokeColor = 'gold';
  brick.fillColor = 'yellow';
}

//Draw a row of n bricks
function drawRow(x, y, n) {
  for(var i = 0; i < n; i++) {
    drawBlock(x + i * BLOCK_SIZE, y);
  }
}

//Draw the rows
var ROW_COUNT = 50;

for(var i = 1; i <= ROW_COUNT; i++) {
  var startX = BLOCK_SIZE / 2 * (ROW_COUNT - i);
  var startY = (i - 1) * BLOCK_SIZE;
  drawRow(startX, startY, i);
}
```

25. Fix Math object bugs

25.1. fix bugs in code

```
function getCoinToss() {
  var r = Math.random();
  if (r < 0.5) {
    return 'heads';
  }
  return 'tails';
}

console.log( getCoinToss() );
console.log( getCoinToss() );
console.log( getCoinToss() );
console.log( getCoinToss() );
console.log( getCoinToss() );
```


25.2. fix bugs in code

```
function getCoinToss() {
  var r = Math.random();
  if (r < 0.5) {
    return 'heads';
  }
  return 'tails';
}

var guess = prompt('Heads or tails?');
guess = guess.toLowerCase();

var toss = getCoinToss();

if (guess != 'heads' && guess != 'tails') {
  alert('Bad input!');
} else if (guess == toss) {
  alert('You Win!');
} else {
  alert('You Lose!');
}
```

25.3. fix bugs in code

```
function getCoinToss() {
  var r = Math.random();
  if (r < 0.5) {
    return 'heads';
  }
  return 'tails';
}

var guess = prompt('Heads or tails?');
guess = guess.toLowerCase();

var toss1 = getCoinToss();
var toss2 = getCoinToss();

var score;

if (guess == toss1 && guess == toss2) {
  score = 2;
} else if (guess == toss1 || guess == toss2) {
  score = 1;
} else {
  score = 0;
}

alert('You scored: ' + score);
```

25.4. fix bugs in code

```
function getCoinToss() {
  var r = Math.random();
  if (r < 0.5) {
    return 'heads';
  }
  return 'tails';
}

var score = 0;

while (true) {
  var guess = prompt('Heads or tails?');
  guess = guess.toLowerCase();

  var toss1 = getCoinToss();
  var toss2 = getCoinToss();

  if (guess == toss1 && guess == toss2) {
    score += 2;
  } else if (guess == toss1 || guess == toss2) {
    score += 1;
  } else {
    break;
  }

  alert('Toss: ' + toss1 + ' ' + toss2 +
    '\nCurrent score: ' + score);
}

alert('Final Score: ' + score);
```

25.5. fix bugs in code

```
function getCoinToss() {
  var r = Math.random();
  if (r < 0.5) {
    return 'heads';
  }
  return 'tails';
}

var score = 0;

while (true) {
  var guess = prompt('Heads or tails?');

  if (guess == null) {
    break;
  }

  guess = guess.toLowerCase();

  var toss1 = getCoinToss();
  var toss2 = getCoinToss();

  if (guess == toss1 && guess == toss2) {
    score += 2;
  } else if (guess == toss1 || guess == toss2) {
    score += 1;
  } else {
    break;
  }

  alert('Toss: ' + toss1 + ' ' + toss2 +
    '\nCurrent score: ' + score);
}

alert('Final Score: ' + score);
```

26. Create a fortune teller

26.1. display a random message

```
var name = prompt('What is your name?');

var msg = 'Hello ' + name + '. ';
var r = Math.random();

if (r < 0.5) {
  msg += 'Nice to meet you.';
} else if (r < 0.7) {
  msg += 'I see your future clearly.';
} else {
  msg += 'Prepare for a surprise!';
}

alert(msg);
```

26.2. predict years to retire

```
var age = prompt('What\'s your age?');
var retire = Math.floor(35 + 41 * Math.random());

var yearsToRetire = retire - age;

if (yearsToRetire > 1) {
  alert('You\'ll retire in ' + yearsToRetire + ' years.');
```


26.3. predict a job

```

var r = Math.random();
console.log(r);
var msg = 'In the near future you\'ll ';

if (r < 0.15) {
  msg += 'start your own software company.';
} else if (r < 0.4) {
  msg += 'work at Google, Facebook or Microsoft.';
} else if (r < 0.95) {
  msg += 'quit learning to code.';
} else {
  msg += 'become a Code Avenger.';
}

alert(msg);

```

26.4. answer a question

```

prompt('Ask me any question about your future.');
```

```

var r = Math.random();
console.log(r);

var msg;

if (r < 0.25) {
  msg = 'No';
} else if (r < 0.4) {
  msg = 'Maybe';
} else if (r < 0.5) {
  msg = 'Possibly';
} else if (r < 0.7) {
  msg = 'Probably';
} else {
  msg = 'Yes';
}

alert(msg);

```

26.5. complete the app

```

var name = prompt('What is your name?')

var msg = 'Hello ' + name + '. ';
var r = Math.random();

if (r < 0.5) {
  msg += 'Nice to meet you.';
} else if (r < 0.7) {
  msg += 'I see your future clearly.';
} else {
  msg += 'Prepare for a surprise!';
}

alert(msg);

var age = prompt('What\'s your age?');
var retire = Math.floor(35 + 41 * Math.random());

var yearsToRetire = retire - age;

if (yearsToRetire > 1) {
  alert('You\'ll retire in ' + yearsToRetire + ' years.');
```

```

} else {
  alert('You\'ll retire soon.');
```

```

}

r = Math.random();
console.log(r);
msg = 'In the near future you\'ll ';

if (r < 0.15) {
  msg += 'start your own software company.';
} else if (r < 0.4) {
  msg += 'work at Google, Facebook or Microsoft.';
} else if (r < 0.95) {
  msg += 'quit learning to code.';
} else {
  msg += 'become a Code Avenger.';
}

alert(msg);

prompt('Ask me any question about your future.');
```

```

r = Math.random();
console.log(r);

if (r < 0.25) {
  msg = 'No';
} else if (r < 0.4) {
  msg = 'Maybe';
} else if (r < 0.5) {
  msg = 'Possibly';
} else if (r < 0.7) {
  msg = 'Probably';
} else {
  msg = 'Yes';
}

alert(msg);

```

27. Refactor graphics code

27.1. replace repeated code with a function

```

//Good code!
function drawBlock(startX, startY, endX, endY) {
  var tl = new Point(startX, startY);
  var br = new Point(endX, endY);
  var block = new Path.Rectangle(tl, br);
  block.strokeWidth = 2;
  block.strokeColor = 'darkGray';
  block.fillColor = 'lightGray';
}

drawBlock(10, 490, 490, 500);
drawBlock(15, 480, 485, 490);
drawBlock(20, 470, 480, 480);

```

27.2. use a function to draw blocks

```

//Draw a block
function drawBlock(startX, startY, endX, endY) {
  var tl = new Point(startX, startY);
  var br = new Point(endX, endY);
  var block = new Path.Rectangle(tl, br);
  block.strokeWidth = 2;
  block.strokeColor = 'darkGray';
  block.fillColor = 'lightGray';
}

//Draw the steps
drawBlock(10, 490, 490, 500);
drawBlock(15, 480, 485, 490);
drawBlock(20, 470, 480, 480);

//Draw 1 pillar
function drawPillar(x) {
  //Draw the pillar
  drawBlock(x + 5, 320, x + 55, 470);
  //Draw the slab on top
  drawBlock(x, 320, x + 60, 315);
}

//Draw all the pillars
for(var i = 0; i < 6; i++) {
  drawPillar(20 + i * 80);
}

```

27.3. use a loop to draw a row of blocks

```

//Draw a block
function drawBlock(startX, startY, endX, endY) {
  var tl = new Point(startX, startY);
  var br = new Point(endX, endY);
  var block = new Path.Rectangle(tl, br);
  block.strokeWidth = 2;
  block.strokeColor = 'darkGray';
  block.fillColor = 'lightGray';
}

//Draw the steps
drawBlock(10, 490, 490, 500);
drawBlock(15, 480, 485, 490);
drawBlock(20, 470, 480, 480);

var PILLAR_COUNT = 8;
var PILLAR_WIDTH = 30;

//Draw 1 pillar
function drawPillar(x) {
  //Draw the pillar
  drawBlock(x + 5, 320, x + PILLAR_WIDTH + 5, 470);
  //Draw the slab on top
  drawBlock(x, 320, x + PILLAR_WIDTH + 10, 315);
}

//Draw all the pillars
for(var i = 0; i < PILLAR_COUNT; i++) {
  drawPillar(20 + i * (PILLAR_WIDTH + 30) );
}

```

27.4. draw several rows of blocks

```

//Draw a block
function drawBlock(startX, startY, endX, endY) {
  var tl = new Point(startX, startY);
  var br = new Point(endX, endY);
  var block = new Path.Rectangle(tl, br);
  block.strokeWidth = 2;
  block.strokeColor = 'darkGray';
  block.fillColor = 'lightGray';
}

var SLAB_HEIGHT = 10;
var SLAB_EDGE = 5;
var PILLAR_TOP = 280;
var PILLAR_BOTTOM = 470;
var PILLAR_COUNT = 8;
var PILLAR_WIDTH = 30;
var PILLAR_GAP = 30;
var PILLAR_LINE_COUNT = 5;
var PILLAR_LINE_GAP = PILLAR_WIDTH / (PILLAR_LINE_COUNT + 1);

//Draw 1 pillar
function drawPillar(x) {
  //Draw the pillar
  drawBlock(x + SLAB_EDGE, PILLAR_TOP, x + PILLAR_WIDTH +
SLAB_EDGE, PILLAR_BOTTOM);

  //Draw lines on the pillar
  for(var i = 1; i <= PILLAR_LINE_COUNT; i++) {
    var p = new Path();
    var x1 = x + SLAB_EDGE + i * PILLAR_LINE_GAP;
    p.add(new Point(x1, PILLAR_TOP));
    p.add(new Point(x1, PILLAR_BOTTOM));
    p.strokeColor = 'silver';
    p.strokeWidth = 2;
  }

  //Draw the slab on top
  drawBlock(x, PILLAR_TOP, x + PILLAR_WIDTH + SLAB_EDGE * 2,
PILLAR_TOP - SLAB_HEIGHT);
}

//Draw all the pillars
for(var i = 0; i < PILLAR_COUNT; i++) {
  drawPillar(20 + i * (PILLAR_WIDTH + PILLAR_GAP) );
}

//Draw the steps
drawBlock(10, 490, 490, 500);
drawBlock(15, 480, 485, 490);
drawBlock(20, 470, 480, 480);

```

27.5. practice drawing shapes

```

//Draw a block
function drawBlock(startX, startY, endX, endY) {
  var tl = new Point(startX, startY);
  var br = new Point(endX, endY);
  var block = new Path.Rectangle(tl, br);
  block.strokeWidth = 2;
  block.strokeColor = 'darkGray';
  block.fillColor = 'lightGray';
}

var SLAB_HEIGHT = 10;
var SLAB_EDGE = 5;
var PILLAR_TOP = 280;
var PILLAR_BOTTOM = 470;
var PILLAR_COUNT = 8;
var PILLAR_WIDTH = 30;
var PILLAR_GAP = 30;
var PILLAR_LINE_COUNT = 5;
var PILLAR_LINE_GAP = PILLAR_WIDTH / (PILLAR_LINE_COUNT + 1);

//Draw 1 pillar
function drawPillar(x) {
  //Draw the pillar
  drawBlock(x + SLAB_EDGE, PILLAR_TOP, x + PILLAR_WIDTH +
SLAB_EDGE, PILLAR_BOTTOM);

  //Draw lines on the pillar
  for(var i = 1; i <= PILLAR_LINE_COUNT; i++) {
    var p = new Path();
    var x1 = x + SLAB_EDGE + i * PILLAR_LINE_GAP;
    p.add(new Point(x1, PILLAR_TOP));
    p.add(new Point(x1, PILLAR_BOTTOM));
    p.strokeColor = 'silver';
    p.strokeWidth = 2;
  }

  //Draw the slab on top
  drawBlock(x, PILLAR_TOP, x + PILLAR_WIDTH + SLAB_EDGE * 2,
PILLAR_TOP - SLAB_HEIGHT);
}

//Draw all the pillars
for(var i = 0; i < PILLAR_COUNT; i++) {
  drawPillar(20 + i * (PILLAR_WIDTH + PILLAR_GAP) );
}

//Draw the steps
drawBlock(10, 490, 490, 500);
drawBlock(15, 480, 485, 490);
drawBlock(20, 470, 480, 480);

//Draw the slabs under the roof
drawBlock(20, 250, 480, 270);
drawBlock(20, 225, 480, 250);
drawBlock(10, 215, 490, 225);

//Draw the roof
function drawTriangle(x, y, width, height) {
  var t = new Path();
  t.add(new Point(x, y));
  t.add(new Point(x + width / 2, y - height));
  t.add(new Point(x + width, y));
  t.strokeWidth = 2;
  t.strokeColor = 'darkGray';
  t.fillColor = 'lightGray';
  t.closed = true;
}

drawTriangle(10, 215, 480, 80);
drawTriangle(30, 215, 440, 70);

```

28. Refactor a dice game called Chuck-A-Luck

28.1. create a function to avoid repeated code

```

function getDiceRoll() {
  return Math.floor(Math.random() * 6 + 1);
}

var roll1 = getDiceRoll();
var roll2 = getDiceRoll();
var roll3 = getDiceRoll();

console.log('Rolls: '
+ roll1 + ' ' + roll2 + ' ' + roll3);

```

28.2. refactor an if statement to remove repeated code

```

function getDiceRoll() {
  return Math.floor(Math.random() * 6 + 1);
}

var guess = prompt('Guess a number: ');
var roll1 = getDiceRoll();
var roll2 = getDiceRoll();
var roll3 = getDiceRoll();

var result = 'lose';
if (roll1 == guess || roll2 == guess || roll3 == guess) {
  result = 'win';
}

alert('Rolls: ' + roll1 + ' ' + roll2 + ' ' + roll3
+ '\nYou ' + result);

```

28.3. refactor by adding a loop

```

function getDiceRoll() {
  return Math.floor(Math.random() * 6 + 1);
}

var guess = prompt('Guess a number: ');
var correct = 0;

for(var i = 0; i < 3; i++) {
  var roll = getDiceRoll();
  if (roll == guess) {
    correct++;
  }
}

alert('You scored ' + correct * 10 + ' points');

```

28.4. refactor by combining if statements

```

function getDiceRoll() {
  return Math.floor(Math.random() * 6 + 1);
}

var guess = prompt('Guess a number: ');
var correct = 0;

for(var i = 0; i < 3; i++) {
  var roll = getDiceRoll();
  if (roll == guess) {
    correct++;
  }
}

var points;

if (correct == 1) {
  points = 10;
} else if (correct == 2) {
  points = 20;
} else if (correct == 3) {
  points = 100;
} else {
  points = -10;
}

alert('You scored ' + points + ' points');

```

28.5. code shortcuts

```
function getDiceRoll() {
  return Math.floor(Math.random() * 6 + 1);
}

// The player starts with 50 points
var totalPoints = 50;

//Keep looping until the player has no more points
while (totalPoints > 0) {
  var guess = prompt('Guess a number: ');
  //If the player clicks cancel then stop
  if (guess == null) {
    break;
  }

  //The rolls variable is a list of all dice rolls
  var rolls = '';
  var correct = 0;

  //Roll 3 dice and count how many are correct
  for(var i = 0; i < 3; i++) {
    var roll = getDiceRoll();
    if (roll == guess) {
      correct++;
    }
    rolls += roll + ' ';
  }

  var points;

  if (correct == 1) {
    points = 10;
  } else if (correct == 2) {
    points = 20;
  } else if (correct == 3) {
    points = 100;
  } else {
    points = -10;
  }

  //Update the players total points
  totalPoints += points;

  alert(rolls + '\nScore: ' + totalPoints + ' points');
}
```

29. Create a number guessing game

29.1. get a random number from 1-100

```
var randomNumber = Math.floor(Math.random() * 100);
console.log(randomNumber);
```

29.2. ask the player to guess

```
var randomNumber = Math.floor(Math.random() * 100);
console.log(randomNumber);

var guess;
while (guess != randomNumber) {
  guess = prompt('Guess the number');
}

alert('Congratulations!');
```

29.3. ending the game early

```
var randomNumber = Math.floor(Math.random() * 100);
console.log(randomNumber);

var guess;
while (guess != randomNumber) {
  guess = prompt('Guess the number');

  if (guess == null || guess == '' || isNaN(guess)) {
    break;
  }

  if (guess == randomNumber) {
    alert('Congratulations!');
  }
}
```

29.4. give hints

```
var randomNumber = Math.floor(Math.random() * 100);
console.log(randomNumber);

var guess;
while (guess != randomNumber) {
  guess = prompt('Guess the number');

  if (guess == null || guess == '' || isNaN(guess)) {
    break;
  }

  if (guess < randomNumber) {
    alert('Higher');
  } else if (guess > randomNumber) {
    alert('Lower');
  }

  if (guess == randomNumber) {
    alert('Congratulations!');
  }
}
```

29.5. 2 player game

```
var randomNumber = Math.floor(Math.random() * 100);
console.log(randomNumber);

var guess;
var whosTurn = 2;
while (guess != randomNumber) {
  whosTurn = 3 - whosTurn;

  guess = prompt('Player ' + whosTurn + ', guess a number');

  if (guess == null || guess == '' || isNaN(guess)) {
    break;
  }

  if (guess < randomNumber) {
    alert('Higher');
  } else if (guess > randomNumber) {
    alert('Lower');
  }

  if (guess == randomNumber) {
    alert('Player ' + whosTurn + ' wins');
  }
}
```

30. Review lessons 21-29

30.1. review rounding functions

Review Quiz Questions:

1. 23
2. 24
3. 23.5
4. 24

30.2. review Math object functions

Review Quiz Questions:

1. 8
2. 9
3. 10
4. 7

30.3. review the random function

Review Quiz Questions:

1. 9
2. 0
3. 8
4. 1

30.4. review the random function

Review Quiz Questions:

1. 14
2. 5
3. 13
4. 6

30.5. review Math object function names

Review Quiz Questions:

1. ceil
2. sqrt
3. pow
4. abs

31. Intro to lessons 32-40

31.1. create a list with separate variables

```
var name1 = 'Nellie W';
var name2 = 'Richard O';
var name3 = 'Aaron M';
var name4 = 'Jenna W';
var name5 = 'Fiona S';
var name6 = 'Robert A';
var name7 = 'Ian W';
var name8 = 'Mike W';

console.log('Class list')
console.log(name1 + ', ' + name2 + ', ' + name3 + ', ' + name4 +
', ' + name5 + ', ' + name6 + ', ' + name7 + ', ' + name8);
```

31.2. create a list with an array

```
var names = ['Nellie W', 'Richard O', 'Aaron M',
'Jenna W', 'Fiona S', 'Robert A', 'Ian W',
'Mike W', 'Adam W'];

console.log('Class list')
console.log(names);
```

31.3. print single items from an array

```
var names = ['Nellie W', 'Richard O', 'Aaron M',
'Jenna W', 'Fiona S', 'Robert A', 'Ian W',
'Mike W', 'Adam W', 'Holly O'];

//Print a List of male names
console.log('Males');
console.log(names[1]);
console.log(names[2]);
console.log(names[5]);
console.log(names[6]);
console.log(names[7]);
console.log(names[8]);

//Print a List of female names
console.log('\nFemales');
console.log(names[0]);
console.log(names[3]);
console.log(names[4]);
console.log(names[9]);
```

31.4. manually sort a list

```
var names = ['Nellie W', 'Richard O', 'Aaron M',
'Jenna W', 'Fiona S', 'Robert A', 'Ian W',
'Mike W', 'Adam W', 'Holly O', 'Chris B'];

console.log('Sorted class list');
console.log(names[2]);
console.log(names[8]);
console.log(names[10]);
console.log(names[4]);
console.log(names[9]);
console.log(names[6]);
console.log(names[3]);
console.log(names[7]);
console.log(names[0]);
console.log(names[1]);
console.log(names[5]);
```

31.5. print sorted list

```
var names = [
'Nellie W', 'Richard O', 'Aaron M', 'Jenna W',
'Fiona S', 'Robert A', 'Ian W', 'Mike W',
'Adam W', 'Holly O', 'Chris B', 'Stuart C'
];

console.log('Unsorted class list');
console.log(names);

console.log('\nSorted class list');
console.log(names.sort());
```

32. Sorting and printing lists of strings

32.1. sort an array and print each item on a separate line

```
var names = [
'Nellie W', 'Richard O', 'Aaron M', 'Jenna W',
'Fiona S', 'Robert A', 'Ian W', 'Mike W',
'Mark S', 'Tony C', 'Alena E', 'Crystal V'
];

names.sort();

console.log('Team 1 ' + names[0]);
console.log('Team 1 ' + names[1]);
console.log('Team 1 ' + names[2]);
console.log('Team 1 ' + names[3]);
console.log('Team 2 ' + names[4]);
console.log('Team 2 ' + names[5]);
console.log('Team 2 ' + names[6]);
console.log('Team 2 ' + names[7]);
console.log('Team 3 ' + names[8]);
console.log('Team 3 ' + names[9]);
console.log('Team 3 ' + names[10]);
console.log('Team 3 ' + names[11]);
```

32.2. use a loop to print items

```
var names = [
'Nellie W', 'Richard O', 'Aaron M', 'Jenna W',
'Fiona S', 'Robert A', 'Ian W', 'Mike W',
'Adam W', 'Holly O', 'Chris B', 'Stuart C',
'Chris B', 'LeBron J', 'Dwayne W', 'Shane B',
'Kobe B', 'Dwight H', 'Steve N', 'Pau G',
'Kevin L', 'Ricky R', 'Andrei K', 'Brandon R',
'Tony P', 'Manu G', 'Tim D', 'Danny G',
'Maria S', 'Anna K', 'Steffi G', 'Venus W'
];

names.sort();

for (var i = 10; i < 25; i++) {
  console.log(names[i]);
}
```

32.3. add team numbers to the list

```

var names = [
  'Nellie W', 'Richard O', 'Aaron M', 'Jenna W',
  'Fiona S', 'Robert A', 'Ian W', 'Mike W',
  'Adam W', 'Holly O', 'Chris B', 'Stuart C',
  'Chris B', 'LeBron J', 'Dwayne W', 'Shane B',
  'Kobe B', 'Dwight H', 'Steve N', 'Pau G',
  'Kevin L', 'Ricky R', 'Andrei K', 'Brandon R',
  'Tony P', 'Manu G', 'Tim D', 'Danny G',
  'Maria S', 'Anna K', 'Steffi G', 'Venus W'
];

names.sort();

var TEAM_SIZE = 4;

for (var i = 0; i < 32; i++) {
  var teamNo = Math.floor(i / TEAM_SIZE + 1);
  console.log('Team ' + teamNo + ' ' + names[i]);
}

```

32.4. get the length of an array

```

var names = [
  'Nellie W', 'Richard O', 'Aaron M', 'Jenna W',
  'Fiona S', 'Robert A', 'Ian W', 'Mike W',
  'Adam W', 'Holly O', 'Chris B', 'Stuart C',
  'Chris B', 'LeBron J', 'Dwayne W', 'Shane B',
  'Kobe B', 'Dwight H', 'Steve N', 'Pau G',
  'Kevin L', 'Ricky R', 'Andrei K', 'Brandon R',
  'Tony P', 'Manu G', 'Tim D', 'Danny G',
  'Maria S', 'Anna K', 'Steffi G', 'Venus W'
];

var name = prompt('Enter a student name:');

while (name != null) {
  names.push(name);
  name = prompt('Enter the next student name:');
}

names.sort();

console.log('Class size: ' + names.length);

console.log(names[0]);
console.log(names[1]);
console.log(names[names.length - 1]);
console.log(names[names.length - 2]);

```

32.5. use length to print an array of unknown length

```

var names = [
  'Nellie W', 'Richard O', 'Aaron M', 'Jenna W',
  'Fiona S', 'Robert A', 'Ian W', 'Mike W',
  'Adam W', 'Holly O', 'Chris B', 'Stuart C',
  'Chris B', 'LeBron J', 'Dwayne W', 'Shane B',
  'Kobe B', 'Dwight H', 'Steve N', 'Pau G',
  'Kevin L', 'Ricky R', 'Andrei K', 'Brandon R',
  'Tony P', 'Manu G', 'Tim D', 'Danny G',
  'Maria S', 'Anna K', 'Steffi G', 'Venus W'
];

var name = prompt('Enter a student name:');

while (name != null) {
  names.push(name);
  name = prompt('Enter the next student name:');
}

names.sort();

var TEAM_SIZE = 8;

for (var i = 0; i < names.length; i++) {
  var teamNo = Math.floor(i / TEAM_SIZE + 1);
  console.log('Team ' + teamNo + ' ' + names[i]);
}

```

33. Practice creating arrays

33.1. use push to add to an array

```

var array = [72, 15, 22, 99];

var array2 = [];

array2.push(72);
array2.push(15);
array2.push(22);
array2.push(99);

console.log(array2);

```

33.2. modify items in an array

```

var array = [72, 15, 22, 99];

array[1] = 150;
array[2] = 220;

console.log(array);

```

33.3. modify undefined items in an array

```

var array = [72, 15, 22, 99];

console.log(array[4]);
array[9] = 44;
console.log(array);

```

33.4. use a loop to print values in an array on separate lines

```

var a;
console.log(a);

var array = [72, 15, 22, 99];
array[9] = 44

for (var i = 0; i < array.length; i++) {
  console.log(array[i]);
}

```

33.5. modifying items in an array by looping

```

//Create an array of even numbers from 2 to 100
var numbers = [];

for(var i = 0; i < 50; i++) {
  numbers[i] = (i + 1) * 2;
}

console.log(numbers);

//Create array with multiples of 4 up to 200
for(var i = 0; i < numbers.length; i++) {
  numbers[i] *= 2;
}

console.log(numbers);

```

34. Refactor arrays with graphics

34.1. move 3 blocks

```

var pegs = [];

for(var i = 0; i < 7; i++) {
  var x = 45 + i * 50;
  var center = new Point(x, 45);
  var c = new Path.Circle(center, 25);
  c.strokeColor = 'black';
  pegs.push(c);
}

```

34.2. replace numbers with constants

```

var pegs = [];
var RADIUS = 25;
var ORIGIN = 45;

for(var i = 0; i < 7; i++) {
  var x = ORIGIN + i * 2 * RADIUS;
  var center = new Point(x, ORIGIN);
  var c = new Path.Circle(center, RADIUS);
  c.strokeColor = 'black';
  pegs.push(c);
}

```

34.3. replace strings with constants

```

var pegs = [];
var RADIUS = 25;
var ORIGIN = 45;

for(var i = 0; i < 7; i++) {
  var x = ORIGIN + i * 2 * RADIUS;
  var center = new Point(x, ORIGIN);
  var c = new Path.Circle(center, RADIUS);
  c.strokeColor = 'black';
  pegs.push(c);
}

var COLOR1 = 'blue';
var COLOR2 = 'red';
var EMPTY = 'white';

pegs[0].fillColor = COLOR1;
pegs[1].fillColor = COLOR1;
pegs[2].fillColor = COLOR1;
pegs[3].fillColor = EMPTY
pegs[4].fillColor = COLOR2;
pegs[5].fillColor = COLOR2;
pegs[6].fillColor = COLOR2;

```

34.4. use a loop to set initial colors

```

var pegs = [];
var RADIUS = 25;
var ORIGIN = 45;

var COLOR1 = 'blue';
var COLOR2 = 'red';
var EMPTY = 'white';
var START_COLORS = [COLOR1, COLOR1, COLOR1,
  EMPTY, COLOR2, COLOR2, COLOR2];

for(var i = 0; i < 7; i++) {
  var x = ORIGIN + i * 2 * RADIUS;
  var center = new Point(x, ORIGIN);
  var c = new Path.Circle(center, RADIUS);
  c.strokeColor = 'black';
  c.fillColor = START_COLORS[i];
  pegs.push(c);
}

```

34.5. draw several rows of blocks

```

var pegs = [];
var RADIUS = 35;
var ORIGIN = 40;

var COLOR1 = 'orange';
var COLOR2 = 'purple';
var EMPTY = 'white';
var START_COLORS = [COLOR1, COLOR1, COLOR1,
  EMPTY, COLOR2, COLOR2, COLOR2];

//Setup the pegs
for(var i = 0; i < 7; i++) {
  var x = ORIGIN + i * 2 * RADIUS;
  var center = new Point(x, ORIGIN);
  var c = new Path.Circle(center, RADIUS);
  c.strokeColor = 'black';
  c.fillColor = START_COLORS[i];
  pegs.push(c);
}

//Dummy item used to check peg color
var item = new Path();

//Checks the color of a peg
function isPeg(no, color) {
  item.fillColor = color;
  return no >= 0 && no < pegs.length &&
    pegs[no].fillColor == item.fillColor;
}

//Move a peg
//from - the index of the peg to move
//to - where to move the peg
function move(from, to) {
  pegs[to].fillColor = pegs[from].fillColor;
  pegs[from].fillColor = EMPTY;
}

//Try to move the peg to 1 of the 2 valid positions
//no - the index of the peg to move
//direction - 1 if the peg should move right, -1 if the peg
should move left
function tryMove(no, direction) {
  if(isPeg(no + direction, EMPTY)) {
    move(no, no + direction);
  } else if (isPeg(no + direction * 2, EMPTY)) {
    move(no, no + direction * 2);
  }
}

//Called by Paper.js each time the mouse is clicked
function onMouseUp(event) {
  //Get the index of the peg that was clicked
  var no = pegs.indexOf(event.item);

  if (no >= 0 && isPeg(no, COLOR1)) {
    tryMove(no, 1);
  } else if (no >= 0 && isPeg(no, COLOR2)) {
    tryMove(no, -1);
  }
}

```

35. Arrays practice

35.1. determine the value elements in an array

35.2. undefined array elements

35.3. array creation practice

35.4. changing arrays with a function

35.5. adding to an array with a function

36. Functions with an array as a parameter

36.1. calculate sum and mean of a list of variables

```

var class1 = [75, 53, 92, 100, 24,
             85, 43, 55, 84, 82];

var class2 = [54, 92, 72, 65, 66,
             83, 44, 81, 88, 78];

function sum(a, b, c, d, e, f, g, h, i, j) {
  return a + b + c + d + e + f + g + h + i + j;
}

function mean(a, b, c, d, e, f, g, h, i, j) {
  return sum(a, b, c, d, e, f, g, h, i, j) / 10;
}

var mean1 = mean(class1[0], class1[1], class1[2], class1[3],
class1[4], class1[5], class1[6], class1[7], class1[8],
class1[9]);

var mean2 = mean(class2[0], class2[1], class2[2], class2[3],
class2[4], class2[5], class2[6], class2[7], class2[8],
class2[9]);

console.log('Class 1 mean: ' + Math.round(mean1));
console.log('Class 2 mean: ' + Math.round(mean2));

```

36.2. create a function with an array as a parameter

```

var class1 = [75, 53, 92, 100, 24, 55, 86,
             85, 43, 55, 84, 82, 99, 51];

var class2 = [54, 92, 72, 65, 66,
             83, 44, 81, 88, 78];

function sum(array) {
  var sum = 0;
  for (var i = 0; i < array.length; i++) {
    sum += array[i];
  }
  return sum;
}

function mean(array) {
  return sum(array) / array.length;
}

var mean1 = mean(class1);
var mean2 = mean(class2);

console.log('Class 1 mean: ' + Math.round(mean1));
console.log('Class 2 mean: ' + Math.round(mean2));

```

36.3. sort an array

```

var class1 = [75, 53, 92, 100, 24, 55, 86,
             85, 43, 8, 84, 82, 100, 51];

var class2 = [54, 92, 9, 65, 66,
             83, 44, 81, 88, 100];

class1.sort();
class2.sort();

console.log(class1);
console.log(class2);

```

36.4. sort an array of numbers

```

var class1 = [75, 53, 92, 100, 24, 55, 86,
             85, 43, 8, 84, 82, 100, 51];

var class2 = [54, 92, 9, 65, 66,
             83, 44, 81, 88, 100];

function compareNumbers(a, b) {
  return a - b;
}

function compareNumbers2(a, b) {
  return b - a;
}

console.log('Before sorting');
console.log(class1);
console.log(class2);

class1.sort(compareNumbers);
class2.sort(compareNumbers2);

console.log('\nAfter sorting');
console.log(class1);
console.log(class2);

```

36.5. calculate the range of an array

```

var class1 = [75, 53, 92, 100, 24, 55, 86,
             85, 43, 8, 84, 82, 100, 51];

var class2 = [54, 92, 9, 65, 66,
             83, 44, 81, 88, 100];

function compareNumbers(a, b) {
  return a - b;
}

function min(array) {
  array.sort(compareNumbers);
  return array[0];
}

function max(array) {
  array.sort(compareNumbers);
  return array[array.length - 1];
}

function range(array) {
  return max(array) - min(array);
}

console.log( range(class1) );
console.log( range(class2) );

```

37. Using the modulus operator

37.1. check if a number is even or odd

37.2. modulus operator

37.3. functions that check even and odd

37.4. using modulus to calculate time

37.5. using modulus to covert times

38. Fix array code

38.1. fix bugs in the code

```
function median(array) {
  //Check if the array length is even
  if (array.length % 2 == 0) {
    //Get the position of the middle items
    var middle1 = array.length / 2
    var middle2 = middle1 - 1;
    //Return the average of those items
    return (array[middle1] + array[middle2]) / 2;
  }

  //If array length is odd return the middle item
  var middleNo = Math.floor(array.length / 2);
  return array[middleNo];
}

var class1 = [14, 35, 39, 66, 74, 93, 94];
var class2 = [7, 25, 45, 72, 79, 83, 91, 100];

console.log(median(class1));
console.log(median(class2));
```

38.2. fix bugs in the code

```
//Create an empty array
var scores = [];

//Ask the user to enter a score
var score = prompt('Enter the first score: ');

//Loop until the input is cancel, or not a number
while(score != null && !isNaN(score)) {
  scores.push(score);
  var score = prompt('Enter the next score: ');
}

console.log(scores);
```

38.3. fix bugs in code

```
function drawAxis() {
  var axis = new Path();
  axis.strokeColor = 'black';
  axis.add(new Point(50, 200));
  axis.add(new Point(50, 450));
  axis.add(new Point(450, 450));
}

drawAxis();
```

38.4. fix bugs in code

```
var ORIGIN_Y = 400;
var ORIGIN_X = 50;
var WIDTH = 400;
var HEIGHT = 300;
var GAP = 0.1;

function drawAxis() {
  var axis = new Path();
  axis.strokeColor = 'black';
  axis.add(new Point(ORIGIN_X, ORIGIN_Y - HEIGHT));
  axis.add(new Point(ORIGIN_X, ORIGIN_Y));
  axis.add(new Point(ORIGIN_X + WIDTH, ORIGIN_Y));
}

function drawBars(array) {
  var sectionWidth = WIDTH / array.length;
  var barWidth = sectionWidth * (1 - 2 * GAP);

  //Draw each bar
  for (var i = 0; i < array.length; i++) {
    //Calculate the location of the bar
    var left = ORIGIN_X + sectionWidth * (i + GAP);
    var right = left + barWidth;
    var top = ORIGIN_Y - array[i] * HEIGHT / 100;
    //Create the bar
    var p1 = new Point(left, ORIGIN_Y);
    var p2 = new Point(right, top);
    var bar = new Path.Rectangle(p1, p2);
    bar.fillColor = 'red';
  }
}

var scores = [14, 35, 39, 66, 74, 93, 94, 100];

drawBars(scores);
drawAxis();
```

38.5. fix bugs in code

```
var ORIGIN_Y = 400;
var ORIGIN_X = 50;
var WIDTH = 400;
var HEIGHT = 300;
var GAP = 0.1;

function drawAxis() {
  var axis = new Path();
  axis.strokeColor = 'black';
  axis.add(new Point(ORIGIN_X, ORIGIN_Y - HEIGHT));
  axis.add(new Point(ORIGIN_X, ORIGIN_Y));
  axis.add(new Point(ORIGIN_X + WIDTH, ORIGIN_Y));
}

function drawBars(array) {
  var sectionWidth = WIDTH / array.length;
  var barWidth = sectionWidth * (1 - 2 * GAP);

  //Draw each bar
  for (var i = 0; i < array.length; i++) {
    //Calculate the location of the bar
    var left = ORIGIN_X + sectionWidth * (i + GAP);
    var right = left + barWidth;
    var top = ORIGIN_Y - array[i] * HEIGHT / 100;
    //Create the bar
    var p1 = new Point(left, ORIGIN_Y);
    var p2 = new Point(right, top);
    var bar = new Path.Rectangle(p1, p2);

    bar.strokeColor = 'black';

    //Alternate colors for the bars
    if (i % 2 == 1) {
      bar.fillColor = 'red';
    } else {
      bar.fillColor = 'blue';
    }
  }
}

var scores = [14, 35, 39, 66, 74, 93, 94, 100];

drawBars(scores);
drawAxis();
```

39. Create a pig latin translator

39.1. check if a letter is a vowel

```
function isVowel (c) {
  c = c.toLowerCase();
  if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c ==
'u') {
    return true;
  }
  return false;
}

console.log(isVowel('a'));
console.log(isVowel('E'));
console.log(isVowel('u'));
console.log(isVowel('Y'));
console.log(isVowel('D'));
console.log(isVowel('z'));
```

39.2. write a function to check if a word is all letters

```
function isVowel (c) {
  c = c.toLowerCase();
  if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c ==
'u') {
    return true;
  }
  return false;
}

function isTranslatable(w) {
  var vowelCount = 0;

  //Loop through each letter in the word
  for (var i = 0; i < w.length; i++) {
    var c = w.charAt(i).toLowerCase();

    if (isVowel(c)) {
      vowelCount++;
    }

    //Check if the character is NOT a letter
    if (c < 'a' || c > 'z') {
      return false;
    }
  }

  return vowelCount > 0;
}

console.log(isTranslatable('virus'));
console.log(isTranslatable('BUG'));
console.log(isTranslatable('wØrm'));
console.log(isTranslatable("Grr"));
```

39.3. write a function that translates a word

```
function isVowel (c) {
  c = c.toLowerCase();
  if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c ==
'u') {
    return true;
  }
  return false;
}

function isTranslatable(w) {
  var vowelCount = 0;

  //Loop through each letter in the word
  for (var i = 0; i < w.length; i++) {
    var c = w.charAt(i).toLowerCase();

    if (isVowel(c)) {
      vowelCount++;
    }

    //Check if the character is NOT a letter
    if (c < 'a' || c > 'z') {
      return false;
    }
  }

  return vowelCount > 0;
}

function translateWord(w) {
  if (isTranslatable(w)) {
    //Search for the 1st vowel
    for (var i = 0; i < w.length; i++) {
      //Check if each letter is a vowel
      if (isVowel(w[i])) {
        break;
      }
    }

    return w.substring(i) + '-' +
      w.substring(0, i) + 'ay';
  }
  return w;
}

console.log(translateWord('virus'));
console.log(translateWord('bug'));
console.log(translateWord('wØRM'));
console.log(translateWord("Grr"));
```

39.4. write a function that translates a sentence to pig latin

```
function isVowel (c) {
  c = c.toLowerCase();
  if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c ==
'u') {
    return true;
  }
  return false;
}

function isTranslatable(w) {
  var vowelCount = 0;

  //Loop through each letter in the word
  for (var i = 0; i < w.length; i++) {
    var c = w.charAt(i).toLowerCase();

    if (isVowel(c)) {
      vowelCount++;
    }

    //Check if the character is NOT a Letter
    if (c < 'a' || c > 'z') {
      return false;
    }
  }

  return vowelCount > 0;
}

function translateWord(w) {
  if (isTranslatable(w)) {
    //Search for the 1st vowel
    for (var i = 0; i < w.length; i++) {
      //Check if each letter is a vowel
      if (isVowel(w.charAt(i))) {
        break;
      }
    }

    return w.substring(i) + '-' +
w.substring(0, i) + 'ay';
  }
  return w;
}

function translateSentence(s) {
  var result = [];
  var words = s.split(' ');

  for(var i = 0; i < words.length; i++) {
    result.push(translateWord(words[i]));
  }

  return result.join(' ');
}

console.log(translateSentence('this is a test'));
console.log(translateSentence('hello world'));
```

39.5. write a function that translates a sentence from pig latin

```
function isVowel (c) {
  c = c.toLowerCase();
  if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c ==
'u') {
    return true;
  }
  return false;
}

function isTranslatable(w) {
  var vowelCount = 0;

  for (var i = 0; i < w.length; i++) {
    var c = w.charAt(i).toLowerCase();
    if ((c < 'a' || c > 'z') && c != '\ ' && c != '-') {
      return false;
    }

    if (isVowel(c)) {
      vowelCount++;
    }
  }

  return vowelCount > 0;
}

function translateWord(w) {
  if (isTranslatable(w)) {
    for (var i = 0; i < w.length; i++) {
      if (isVowel(w.charAt(i))) {
        break;
      }
    }
    return w.substring(i) + '-' + w.substring(0,i) + 'ay';
  }
  return w;
}

function translateSentence(s) {
  var result = [];
  var words = s.split(' ');

  for (var i = 0; i < words.length; i++) {
    result.push(translateWord(words[i]));
  }

  return result.join(' ');
}

function untranslateWord(w) {
  var parts = w.split('-');

  if (parts.length == 1) {
    return w;
  }

  return parts[1].substring(0, parts[1].length - 2) +
parts[0];
}

function untranslateSentence(s) {
  var result = [];
  var words = s.split(' ');

  for (var i = 0; i < words.length; i++) {
    result.push(untranslateWord(words[i]));
  }

  return result.join(' ');
}

console.log(untranslateSentence('est-tay at-ay oon-spay'));
```

40. Review lessons 31-39

40.1. review creating, accessing and modifying arrays

Review Quiz Questions:

1. 4
2. 8
3. undefined
4. undefined

40.2. review using loops to add to arrays

Review Quiz Questions:

1. 0
2. 4
3. 2
4. undefined

40.3. review the modulus operator

Review Quiz Questions:

1. 6
2. 8
3. 0
4. 8

40.4. review modulus functions

Review Quiz Questions:

1. isEven
2. isOdd
3. isMultipleOf3
4. notMultipleOf3

40.5. review array modification

Review Quiz Questions:

1. 1
2. 2
3. 3
4. 3