

color property

sets the color of text.

Example code:

```
<style>
section {
  color: XXX;
}
</style>

<section>
  Example content.
</section>
```

<a> (a) tag

creates a link to another web page.

href gives the address of the web page.

target is set to `"_blank"` to show the linked page in a new tab when the user clicks it.

Example code:

```
<a target="_blank" href="http://google.com">Google</a>

<br>

<a href="/m/about.html">
  About Us</a>
```

area

defines an area inside an image-`{map}`.

coords specifies the coordinates of the area.

href specifies the hyperlink target for the area.

shape specifies the shape of the area.

Example code:

```


<map name="map">
  <area shape="circle" coords="24, 79, 2" title="mercury">
  <area shape="circle" coords="78, 79, 3" title="venus">
  <area shape="circle" coords="96, 79, 3" title="Earth">
  <area shape="circle" coords="112, 79, 3" title="mars">
  <area shape="circle" coords="143, 79, 17.5"
title="jupiter">
  <area shape="circle" coords="196, 79, 15" title="saturn">
  <area shape="circle" coords="230, 79, 7" title="Uranus">
  <area shape="circle" coords="253, 79, 7" title="neptune">
  <area shape="circle" coords="264, 125, 3" title="pluto">
  <area shape="poly" coords="29, 17, 43, 47, 50, 77, 51, 109,
45, 146, 37, 171, 24, 215, 0, 215, 0, 0, 18, 3" title="sun">
</map>
```


 (br) tag

puts text on separate lines.

Example code:

```
Name: Super Cutey<br>
Age: 18 months<br>
Weight: 11kg<br>
Height: 62cm
```

<canvas> (canvas) tag

gives you a way to draw graphics using JavaScript on a webpage.

id Makes it easy to link each canvas to JS and CSS.

height Sets vertical height of canvas

width Sets horizontal width of canvas

Example code:

```
<style>
html{
  background-color:lightGrey;}
#drawing{
  background-color:aliceBlue;
  border: slateGrey dashed 1px;
}
</style>

<canvas id="drawing" width="200" height="200">
  HTML5 Canvas isn't working if you see this text.
</canvas>

<script>
var draw = document.getElementById('drawing');
var rectangle = draw.getContext('2d');
rectangle.rect(25,25,50,50);
rectangle.stroke();
</script>
```

<input> (input) tag

adds an input field where the user enters data.

type sets the type of input.

value sets the text of the input.

checked pre-selects the input when the page loads.

maxlength sets the maximum number of characters that can be entered.

name sets the name and tells the form to send this input to the server when the form is submitted.

Example code:

```
<input type="radio" checked>Radio Button<br>
<input type="checkbox" checked>Checkbox<br>
<input type="color"><br>
<input type="range"><br>
<input type="time"><br>
<input type="week"><br>
<input type="month"><br>
<input type="date" name="birthday"><br>
<input type="datetime-local"><br>
<input type="hidden"><br>
<input type="search"><br>
<input type="number" value="1234"><br>
<input type="text" maxlength="2"><br>
<input type="password" value="text"><br>
<input type="button" value="Button"><br>
<input type="file"><br>
```

input number

adds an input field where the user enters a number.
min sets the minimum number.
max sets the maximum number.
step sets the size of the intervals between legal numbers.

Example code:

```
<style>
input:invalid {
  background-color: red;
}
</style>
<input id="min" type="number" value="1" min="0">min<br>
<input id="max" type="number" value="1.5" max="5">max<br>
<input id="minMax" type="number" value="5.2" min="5"
max="10">min and max<br>
<input id="step" type="number" value="1.07"
step="0.01">step<br>
<button id="numberButton">Show Number Values</button>
<script>
  function showValues() {
    alert(min.value + '\n' + max.value + '\n' + minMax.value
+ '\n'
    + step.value);
  }

  numberButton.onclick = showValues;
</script>
```

*

multiplies numbers
 E.g. 4 * 8 equals 32

Example code:

```
alert(8 * 4);

//Do * before + and -
alert(8 * 4 - 2 * 4);
```

+

adds numbers and joins strings
 4 + 8 equals 12
 '4' + '8', 4 + '8' and '4' + 8 are equal to '48'

Example code:

```
//Add numbers
alert(8 + 4);

//Join strings
alert('8' + '4');
alert('8' + 4);
alert(8 + '4');
```

-

subtracts numbers, e.g. 32 - 24 equals 8

Example code:

```
//Subtract numbers
alert(8 - 4);

//Subtract strings
alert('8' - '4');
alert('8' - 4);
alert(8 - '4');
```

/

divides numbers
 E.g. 16 / 2 equals 8

Example code:

```
alert(16 / 2);

//Do / before + and -
alert(8 + 4 / 2 + 1);
```

Array.length

property that stores the length of the array.
 E.g. {[1, 2, 3, 4].length} is "4"

Example code:

```
var array = [1, 2, 3];
console.log(array.length);
array.push(4, 5, 6);
console.log(array.length);
```

if

Example code:

```
var age = prompt('How old are you?', '21');

if (age <= 12) {
  alert('You are a child');
} else if (age < 20) {
  alert('You are a teen');
} else if (age < 0 || age > 125) {
  alert('Invalid age');
} else {
  alert('You are an adult');
}
```

Math.PI

Represents the ratio of the circumference of a circle to its diameter, i.e. approximately 3.14159

Example code:

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

Math.random()

Returns a random floating point (decimal) number from 0 (inclusive) up to but not including 1, which you can scale to a desired range

Example code:

```
//Get a decimal no. from 0 up to but excluding 1
console.log(Math.random());

//Get a random no. between 0 to 99
console.log(Math.floor(Math.random() * 100));

//Get an even no. up to but not including 100
var evenNo = Math.floor((Math.random() * 50) * 2);
console.log(randomEven);

//Get a random no. from 1 to 6
var diceRoll = Math.floor(Math.random() * 6 + 1);
console.log(diceRoll);
```

Event.item

gets the item at the position of the mouse when the event occurred. It will contain `{null}` if there was no item there.

Example code:

```
var a = new Shape.Circle(40, 40, 20);
a.fillColor = 'black';
var b = new Shape.Circle(140, 90, 20);
b.fillColor = 'black';
var c = new Shape.Circle(40, 140, 20);
c.fillColor = 'black';

function onMouseDown(event) {
  if (event.item) {
    event.item.fillColor = Color.random();
  }
}
```

**

To the power of/exponent. Multiplies a number by itself a given number of times.

Example code:

```
print(2 ** 3)
print(3 ** 4)
```

in

Keyword used in conditional and loop statements to check if a value is in a set of values, a string, a list or a range of numbers.

Example code:

```
#Checks if the letter A is in the word entered by the user
word = input("Enter a word: ").strip().lower()

if "a" in word:
  print("There is an a!")

#Loops for each number in the range
for i in range(0, 11):
  print(i)

#Loops for each word in the list
for color in ["red", "green", "blue"]:
  print(color)

#Checks if the user's input is in the list
day = input("Enter a weekday: ")

if day in ["monday", "tuesday", "wednesday", "thursday",
"friday"]:
  print("Yes, that is a weekday!")
else:
  print("No that's not a weekday!")
```

print(message)

Outputs information to the screen.

message: the information to be printed to the screen
The message could be a string, number, Boolean, calculation or combination of all of these

Example code:

```
#Print a string:
print("Hello, world!")

#Print a number:
print(9)

#Print a variable:
x = "apple"
print(x)

#Print a calculation:
print(3 * 7)

#Print a Boolean expression
print(3 > 7 or 3 < 4)
```

string.format(values)

Inserts values into `{}` placeholders in a string

values: the values to be inserted into the placeholders in the string
There should be the same number of values as placeholders (separated by commas), or placeholders should be numbered based on which values should be inserted.

Example code:

```
#Printing a calculation with the solution
print("3 * 2 + 4 / 7 = {}".format(3 * 2 + 4 / 7))

#Printing a calculation to decimal places .:1f
calculation = 10/3
print("10 divided by 3 = {:.2f}".format(calculation))

calculation = 4
print("4 to two decimal places = {:.2f}".format(calculation))

#Using more than one placeholder/value
print("There are {} kinds of people in the world; those who know {} and those who {}".format(10, "binary", "don't"))

#Using the same value in more than one placeholder (values are numbered from 0)
print("I see {0}, I see {0}, I see {0}!".format("red"))

#Using values out of order
print("My {1} is full of {0}".format("eels", "hovercraft"))

#Using input and variables
name = input("What is your name?")
age = input("How old are you?")

print("You are {} and you are {} years old.".format(name, age))
```

try/except

A structure that allows a block of code that might crash the program by causing an error message (exception) to be tried. If the code in the try block is run successfully then the except block is not run, otherwise the except block is run. This can be used inside a loop to repeat the code until it does work, if user input is involved.

Example code:

```
#Forcing the user to enter a valid number
while True:
  try:
    number = float(input("Enter your height in metres: "))
    break
  except ValueError:
    print("That is not a valid number, please enter your height in metres e.g. 1.5")
```

turtle.end_fill()

Goes just after the last line of code that draws the final part of a shape to be filled.

Example code:

```
import turtle
tiny = turtle.Turtle()

tiny.fillcolor("yellow")

#Fill a square
tiny.begin_fill() #Goes before the shape is started

for i in range(4): #Draws the square
    tiny.forward(100)
    tiny.left(90)

tiny.end_fill() #Goes after it is completed
```