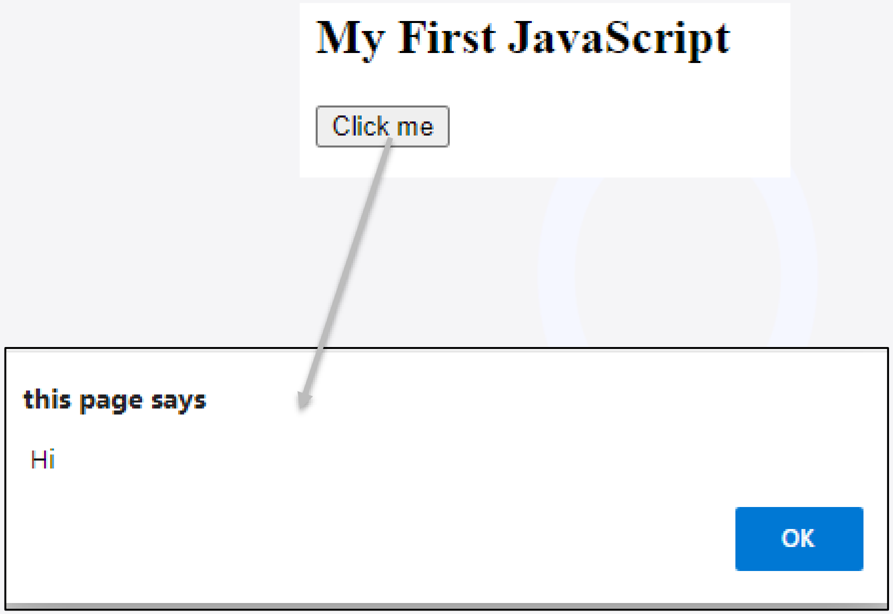
CSIT884:
Web Development

JavaScript Basics

School of Computing and Information Technology
University of Wollongong

My First JavaScript

```
<html>
<body>
<h2>My First JavaScript</h2>
<button onClick="alert('Hi');">
Click me
</button>
</body>
</html>
```



My First JavaScript

this page says

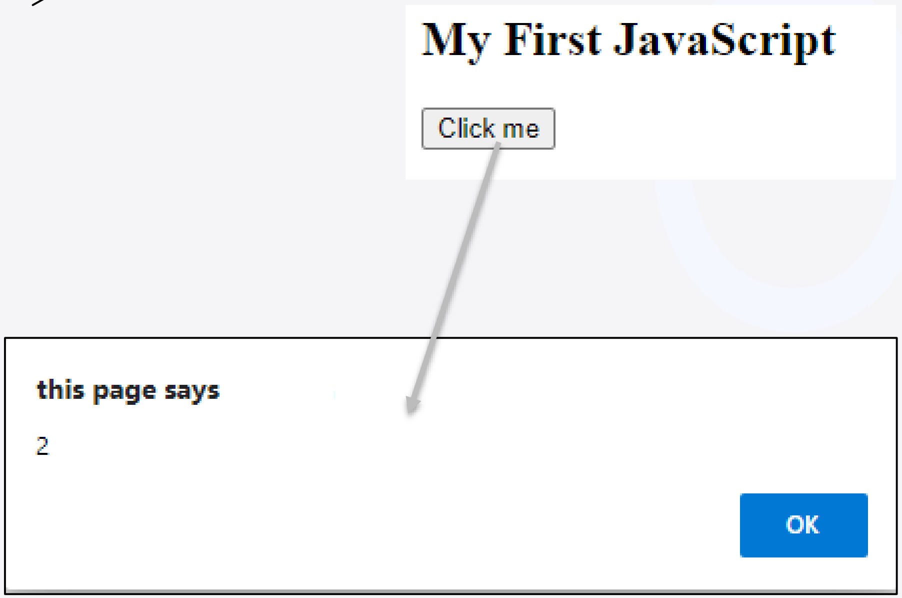
Hi

What will be the outcome if we change it to:

```
<button onClick="alert(1+1);">
Click me
</button>
```

My First JavaScript

```
<button onClick="alert(1+1);">

Click me

</button>
```



My First JavaScript

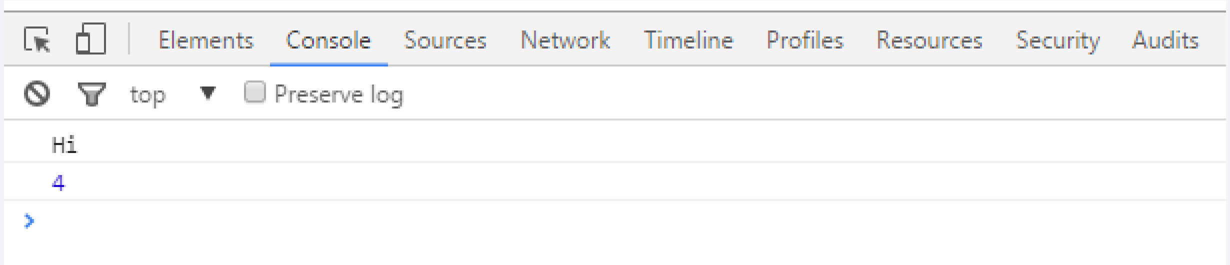
this page says

2

My First JavaScript

```
<button onClick="console.log('Hi');">
Click me
</button>
```

```
<button onClick="console.log(2+2);">
Click me
</button>
```



```
Hi
4
>
```

My First JavaScript

```
<button onClick="alert('Hi'); console.log(2+2);">
Click me
</button>
```

We better put the code inside a function to make it clearer!

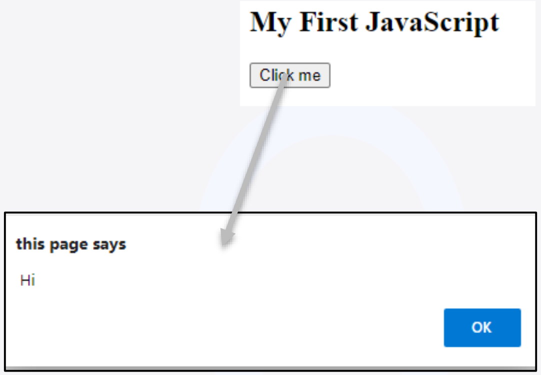
My First JavaScript

```
<button onClick="sayHi();">
Click me
</button>


<script>
function sayHi(){
  alert("Hi");
}
</script>
```

function name

may include multiple arguments separated by commas

code to execute



My First JavaScript

this page says

Hi

Where to include JavaScript

We can put JavaScript code anywhere in the HTML file.

Common practice:

- In the head
- At the end of body

```
<script>

function sayHi(){

  alert("Hi");

}

</script>
```

Where to include JavaScript

In the head

```
<head>

<title>JavaScript Example</title>

<script>

function sayHi(){

  alert("Hi");

}

</script>

</head>
```

Where to include JavaScript

At the end of body (just before the closing body tag)

```
...
<script>
function sayHi(){
  alert("Hi");
}
</script>
</body>
</html>
```

External JavaScript

Instead of putting javascript code inside the html file

```
<script>

function sayHi(){

  alert("Hi");

}

</script>
```

we can specify an external javascript file:

```
<script type="text/javascript" src="js/myscript.js"></script>
```

Basic JavaScript syntax

JavaScript statements are separated by semicolons

```
function silly(){
  alert('Hi’);  - - - - - -  (1)
  console.log(2+2); - - - - - (2)
}
```

Basic JavaScript syntax

JavaScript Comments

Code after double slashes // or between /* and */ is treated as a comment.

Comments are ignored, and will not be executed.

```
/*
this function does a few silly things
 */
function silly(){
  // display an alert box
  alert('Hi');

  // print out the number 4 on the console
  console.log(2+2);
}
```

Basic JavaScript syntax

JavaScript uses the `var` keyword to declare variables.

```
var studentName = "John";

var x, y;

x = 5;

y = x + 2;
```

All JavaScript identifiers are **case sensitive**.

- The variables studentName and StudentName are two different variables.
- The variables x and X are two different variables.

Basic JavaScript syntax

Variable naming: two common conventions

underscore:

student_name, student_id, first_name, last_name

camel case:

studentName, studentId, firstName, lastName

Basic JavaScript syntax

!!!

JavaScript has dynamic types.

This means that the same variable can be used to hold **different data types**:

```
var x;                 // x is undefined
alert(x);

var x = 2016;          // x is a number
alert(x);

var x = "Wollongong";  // x is a string
alert(x);
```

A variable declared without a value will have the value **undefined**.

Basic JavaScript syntax

JavaScript data type: number

```
var age = 19;
var pi = 3.14;
```

Arithmetic operators are used to perform arithmetic on numbers

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus

Basic JavaScript syntax

JavaScript data type: string

```
var age = "19";

var name = 'John';
```

Basic JavaScript syntax

Strings are text, written within double or single quotes:

```
var firstName, lastName, fullName;

firstName = "John";        // using double quotes
lastName = 'Lee';          // using single quotes

fullName = firstName + " " + lastName;
alert(fullName);
```

Use + for string concatenation

Basic JavaScript syntax

Mixing between double or single quotes:

```
var x;

x = "I'm John";          //single quote inside double quotes
alert(x);


x = "My name is 'John'"; //single quotes inside double quotes
alert(x);


x = 'My name is "John"'; //double quotes inside single quotes
alert(x);
```

Basic JavaScript syntax

Change string to number

```
var ageString = "19";

var age = Number(ageString); // age is the number 19
```

Change number to string

```
var age = 19;

var ageString = age.toString(); // ageString is the string "19"
```

Basic JavaScript syntax

JavaScript evaluates expressions from left to right

```
var x;

x = 2016 + "Wollongong";       //2016Wollongong
alert(x);


x = 2016 + 1 + "Wollongong";   //2017Wollongong
alert(x);


x = "Wollongong" + 2016;       //Wollongong2016
alert(x);


x = "Wollongong" + 2016 + 1;   //Wollongong20161
alert(x);
```

Basic JavaScript syntax

JavaScript data type: boolean

```
var authenticated = false;
var isReturningUser = true;
```

```
var x = 5;
var positive = (x > 0);    //true

if(positive){
  alert("x is positive");
}
```

Basic JavaScript syntax

Comparison and Logical Operators

`==`	equal to
`!=`	not equal
`>`	greater than
`<`	less than
`>=`	greater than or equal to
`<=`	less than or equal to
`&&`	logical and
`||`	logical or
`!`	logical not

Basic JavaScript syntax

```
var x = 5;

var y = 6;

if(x == y){

  alert("x and y are equal");

}else{

  alert("x and y are NOT equal");

}
```

if/else statements

……….(1)

```
var x = 5;

var y = 6;

if(x != y){

  alert("x and y are not equal");

}else{

  alert("x and y are equal");

}
```

if/else statements

……….(2)

Basic JavaScript syntax

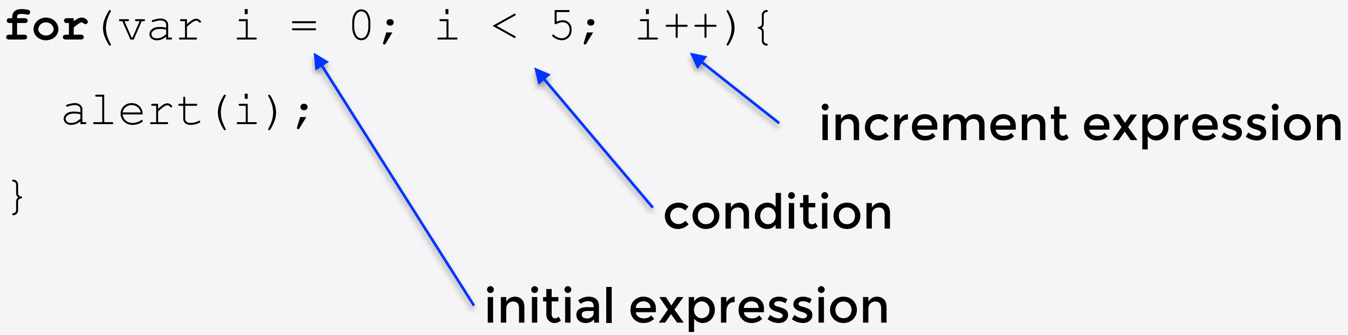
```
var mark = 75;

if(mark > 85){
  alert("Grade A");
}else if (mark > 65){
  alert("Grade B");
}else if (mark > 50){
  alert("Grade C");
}else {
  alert("Grade D");
}
```

Basic JavaScript syntax

```
var day = 5;

switch (day) {
  case 1:
    alert("Monday");
    break;
  case 2:
    alert("Tuesday");
    break;
  case 3:
    alert("Wednesday");
    break;
  case 4:
    alert("Thursday");
    break;
  case 5:
    alert("Friday");
    break;
  case 6:
    alert("Saturday");
    break;
  case 7:
    alert("Sunday");
    break;
  default:
  alert("not in range");
}
```

Basic JavaScript syntax

For-Loop statement:



```
for(var i = 0; i < 5; i++){
  alert(i);
}
```

increment expression

condition

initial expression

Useful tags for dynamic content:

- The `<div>` tag defines a generic section container
- The `<span>` tag defines a generic inline container

Creating dynamic content with JavaScript

- **Step 1:** give the HTML element that we want to change an **ID**

- **Step 2:** use the function

the id of the element

```
var e = document.getElementById("the-id");
```

to get the HTML element that we want to change

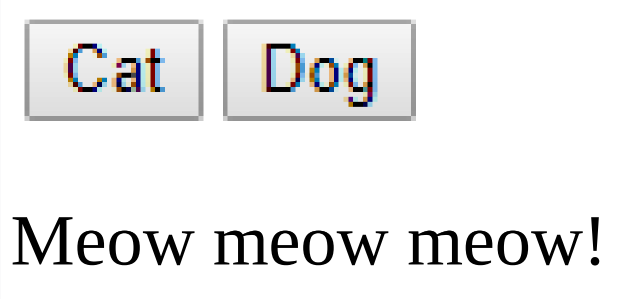
- **Step 3:** change the content of the HTML element

```
    for span, div, etc.:
    e.innerHTML = "the-new-content";

    for input text field:
    e.value = "the-new-value";

    for image:
    e.src = "the-new-image-src";
```

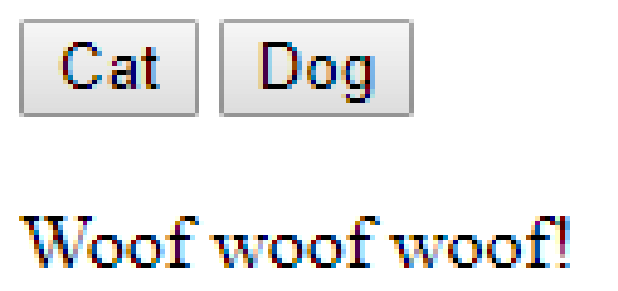
Cat & Dog 1

The web page displays **2 buttons**: “Cat” and “Dog”.

If the user clicks the “Cat” button, a meow-meow message is displayed, and if the user clicks the “Dog” button, a woof-woof message is displayed.



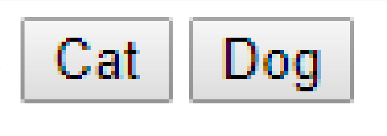
Meow meow meow!



Woof woof woof!

Cat & Dog 1

```
<button onClick="cat()">Cat</button>
<button onClick="dog()">Dog</button>
<br /> <br />
<span id="display"></span>
```

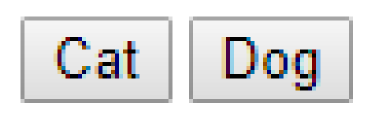


Cat & Dog 1

```
function dog(){
  // get the span element


  // show dog message


}
```

Cat & Dog 1

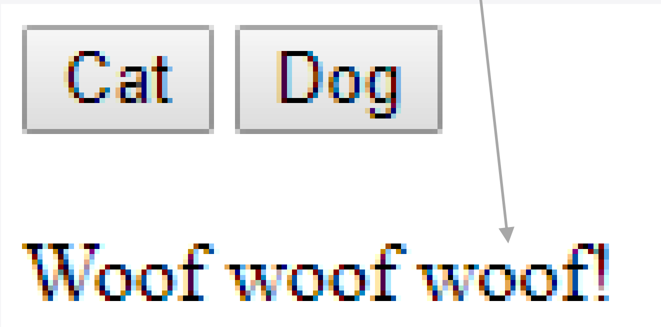
```
function dog(){
  // get the span element
  var displaySpan = document.getElementById("display");


  // show dog message


}
```



```
<span id="display"></span>
```

Cat & Dog 1

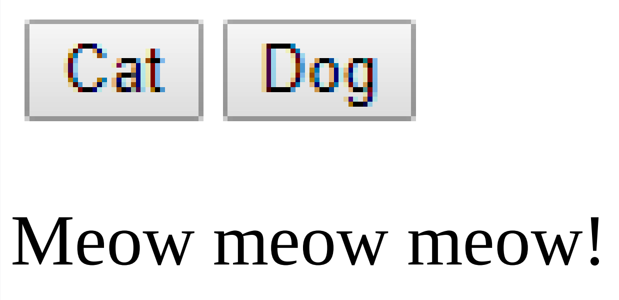
```
function dog(){
  // get the span element
  var displaySpan = document.getElementById("display");

  // show dog message
  displaySpan.innerHTML = "Woof woof woof!";
}
```



Woof woof woof!

Cat & Dog 1

```
function cat(){
  // get the span element
  var displaySpan = document.getElementById("display");

  // show cat message
  displaySpan.innerHTML = "Meow meow meow!";
}
```



Meow meow meow!

That’s not fair!
Why did the dog get to go first?

Creating dynamic content with JavaScript

- **Step 1:** give the HTML element that we want to change an **ID**

- **Step 2:** use the function

```
var e = document.getElementById("the-id");
```

to get the HTML element that we want to change

- **Step 3:** change the content of the HTML element

```
for span, div, etc.:
e.innerHTML = "the-new-content";
```

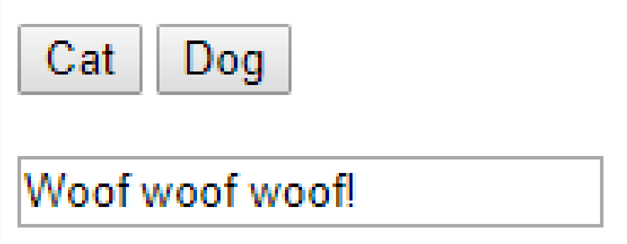
```
for input text field:
e.value = "the-new-value";
```

```
for image:
e.src = "the-new-image-src";
```

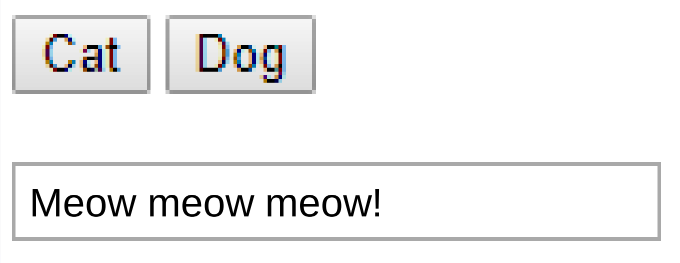
Cat & Dog 2

The web page displays **2 buttons**: “Cat” and “Dog”, and **a text field**.

If the user clicks the “Cat” button, a meow-meow message is displayed inside a text field, and if the user clicks the “Dog” button, a woof-woof message is displayed in a text field.



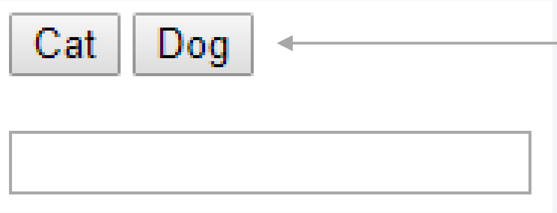
Woof woof woof!



Meow meow meow!

Cat & Dog 2

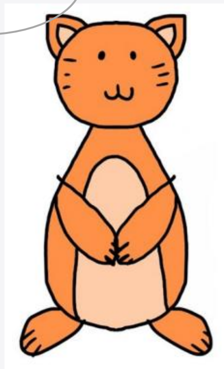
```
<button onClick="cat()">Cat</button>
<button onClick="dog()">Dog</button>
<br /> <br />
<input type="text" id="display" />
```



Cat & Dog 2

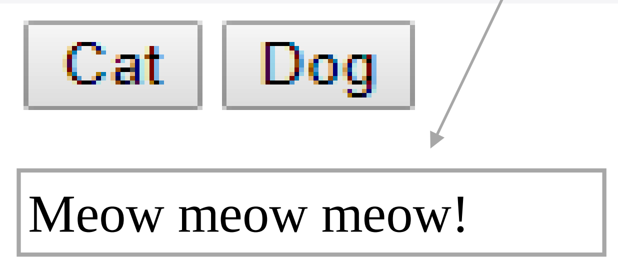
```
function cat(){
  // get the text field element


  // show cat message


}
```

Okay, the cat can go first!



Cat & Dog 2

```
function cat(){
  // get the text field element
  var displayField = document.getElementById("display");

  // show cat message
  displayField.value = "Meow meow meow!";
}
```

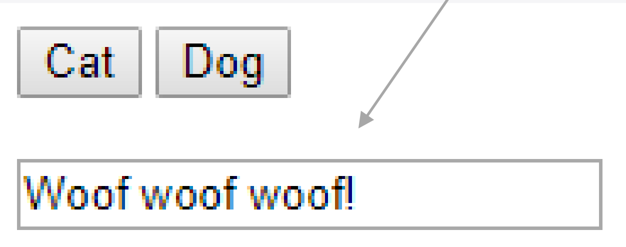


Meow meow meow!

```
(<input type="text" id="display" />)
```

Cat & Dog 2

```
function dog(){
  // get the text field element
  var displayField = document.getElementById("display");

  // show cat message
  displayField.value = "Woof woof woof!";
}
```



Woof woof woof!

```
(<input type="text" id="display" />)
```

Creating dynamic content with JavaScript

- **Step 1:** give the HTML element that we want to change an **ID**

- **Step 2:** use the function

```
var e = document.getElementById("the-id");
```

to get the HTML element that we want to change

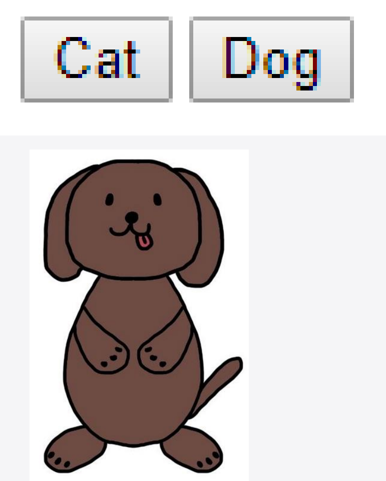
- **Step 3:** change the content of the HTML element

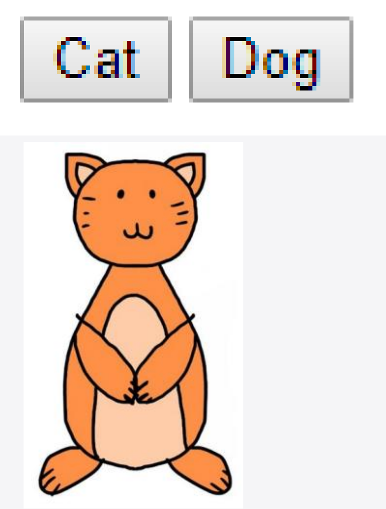
```
for span, div, etc.:
e.innerHTML = "the-new-content";

for input text field:
e.value = "the-new-value";

for image:
e.src = "the-new-image-src";
```

Cat & Dog 3

The web page displays **2 buttons**: “Cat” and “Dog”.

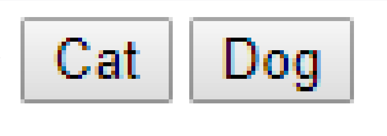
If the user clicks the “Cat” button, an image of a cat is displayed, and if the user clicks the “Dog” button, an image of a dog is displayed.





Cat & Dog 3

```
<button onClick="cat()">Cat</button>

<button onClick="dog()">Dog</button>

<br /> <br />

<img id="display" />
```



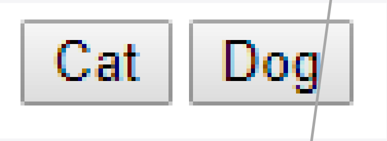
(empty image: no src)

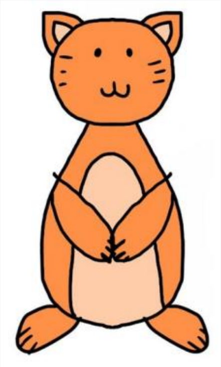
Cat & Dog 3

```
function cat(){
  // get the image element

  // show cat picture
}
```

Cat & Dog 3

```
function cat(){
  // get the image element
  var image = document.getElementById("display");

  // show cat picture
  image.src = "cat.png";
}
```

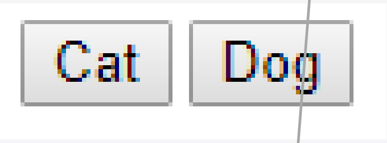


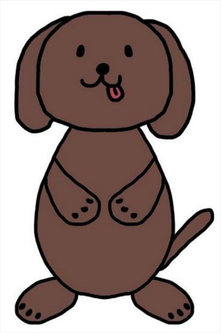


```
<img id="display" />
```

Cat & Dog 3

```
function dog(){
  // get the image element
  var image = document.getElementById("display");

  // show dog picture
  image.src = "dog.png";
}
```





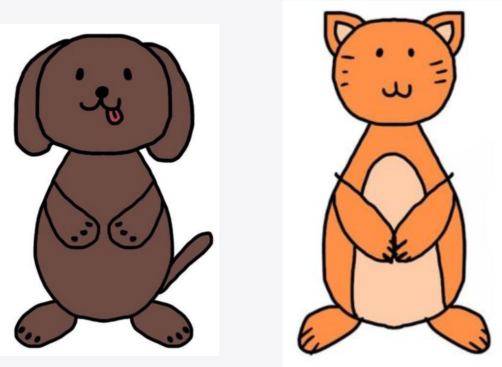
```
<img id="display" />
```

Using variables to save state information

Sometimes we use variables to save the **current status** of the page.

Cat & Dog 4

The web page displays **2 images**: “Cat” and “Dog”, and **2 click counters**.
If the user clicks the “Cat” image, then the click counter for cat is increased.
If the user clicks the “Dog” image, then the click counter for dog is increased.

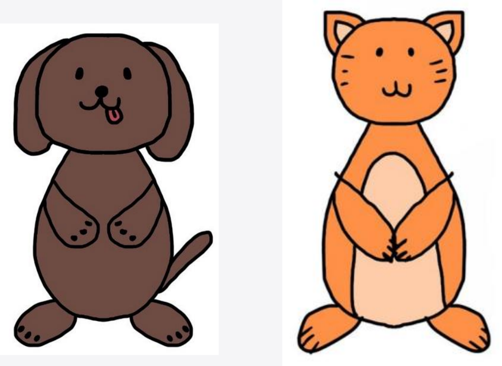


Dog click count: 3

Cat click count: 7

Cat & Dog 4

```
<img src="dog.png" onClick="dog()" />
<img src="cat.png" onClick="cat()" />
<br /> <br />
Dog click count: <span id="dogDisplay">0</span>
<br /> <br />
Cat click count: <span id="catDisplay">0</span>
```



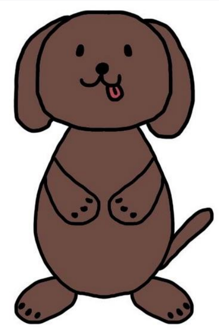
Dog click count: 0

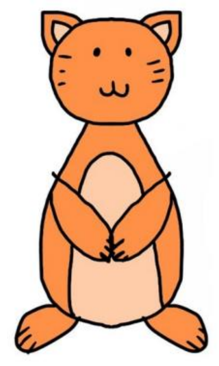
Cat click count: 0

Cat & Dog 4

```
// variable to save the number of dog clicks
var dogClick = 0;
// variable to save the number of cat clicks
var catClick = 0;
```

We use **variables** to save the current number of **dog-clicks** and **cat-clicks**.



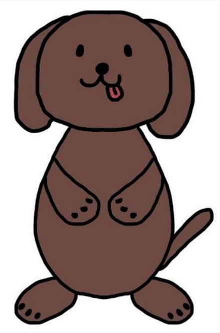


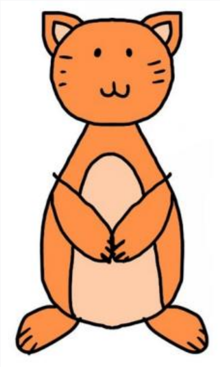
Dog click count: 0

Cat click count: 0

Cat & Dog 4

```
// variable to save the number of dog clicks
var dogClick = 0;

function dog(){
  // increase the number of dog clicks by 1


  // display the number of dog clicks
}
```



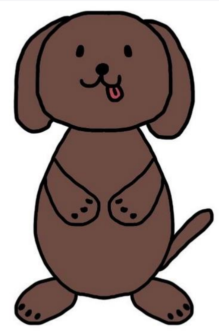


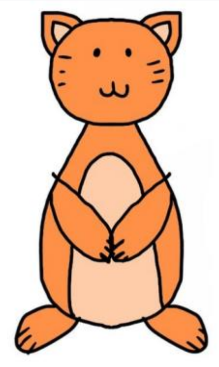
Dog click count: 0

Cat click count: 0

Cat & Dog 4

```
// variable to save the number of dog clicks
var dogClick = 0;

function dog(){
  // increase the number of dog clicks by 1
  dogClick = dogClick + 1;

  // display the number of dog clicks
  var dogSpan = document.getElementById("dogDisplay");
  dogSpan.innerHTML = dogClick;
}
```





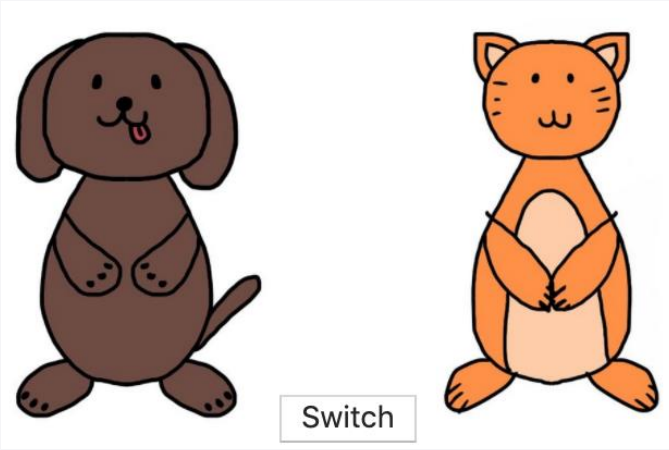
```
<span id="dogDisplay">0</span>
```

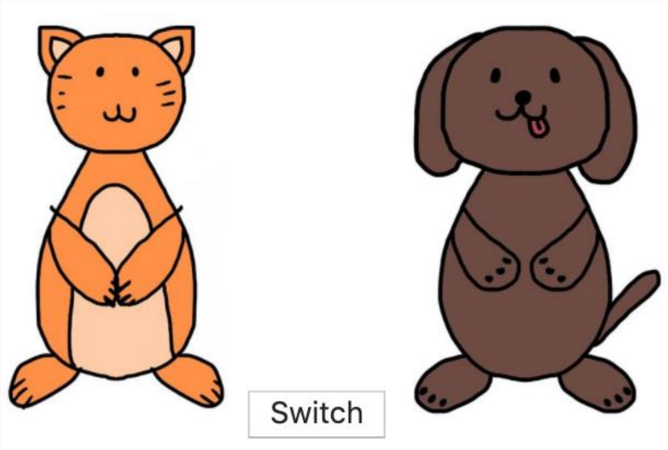
Dog click count: 0

Cat click count: 0

Cat & Dog 5

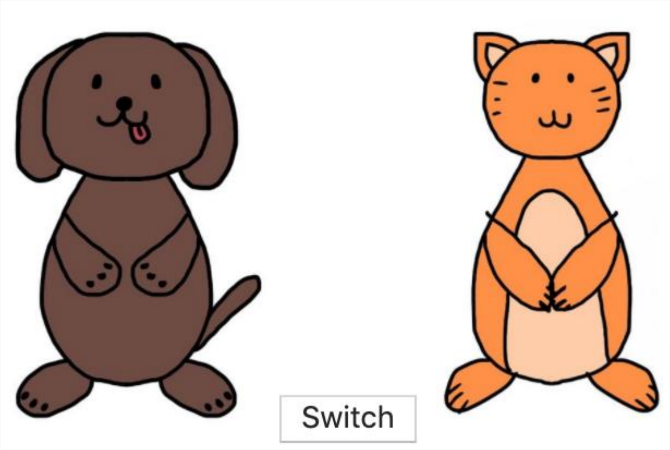
The web page displays **2 images**: “Dog” on the **left**, “Cat” on the **right**, and a **button** “Switch”.
If the user clicks the “Switch” button, then the two images switch their places.





Cat & Dog 5

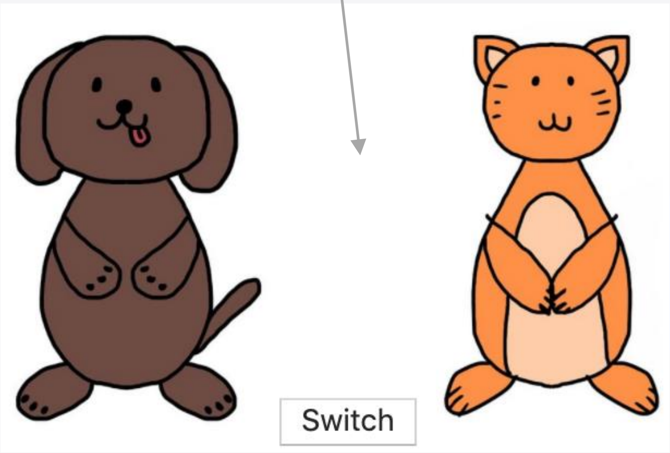
```
<img id="left" src="dog.png" />
<button onClick="switchImage()">
Switch
</button>
<img id="right" src="cat.png" />
```



Cat & Dog 5

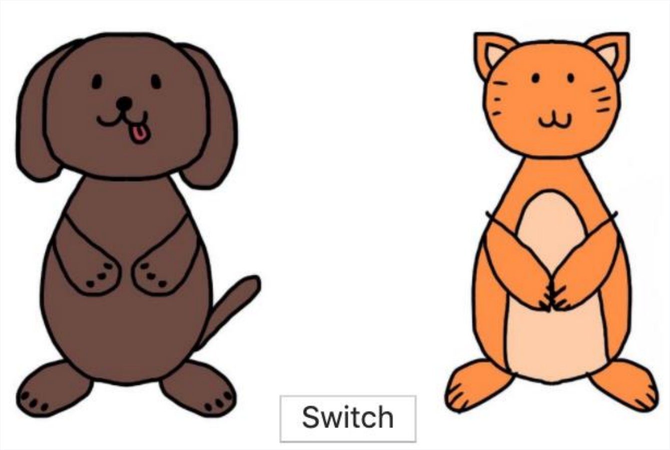
```
// variable to save the position of dog and cat images
// two values: "dog-cat" or "cat-dog"
// original position is "dog-cat"
var position = "dog-cat";
```

We use a **variable** to save the current position of the images



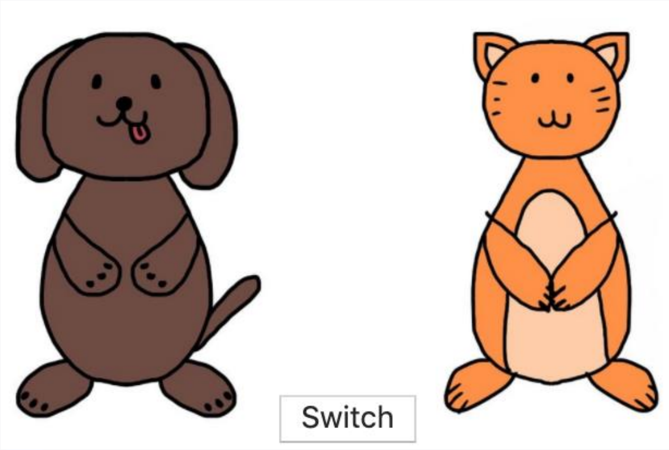
Cat & Dog 5

```
var position = "dog-cat";
function switchImage(){
  // check what is the current position, then switch it
  // change position variable
  // change the images
}
```



Cat & Dog 5

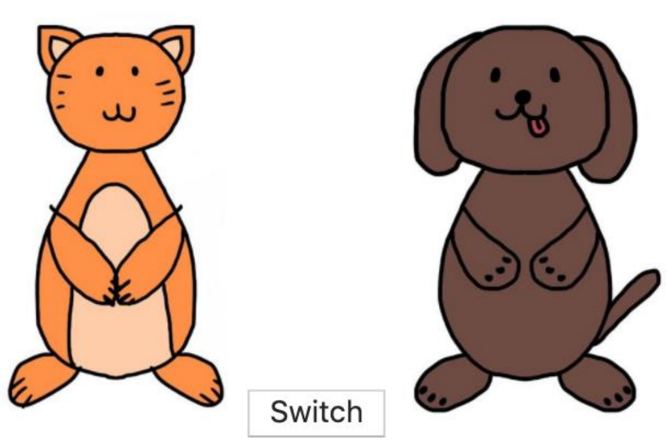
```
if(position == "dog-cat"){
  // change position variable
  position = "cat-dog";

  // change the images
  var leftImage = document.getElementById("left");
  leftImage.src = "cat.png";

  var rightImage = document.getElementById("right");
  rightImage.src = "dog.png";
}else...
```



Current position is dog-cat

Cat & Dog 5

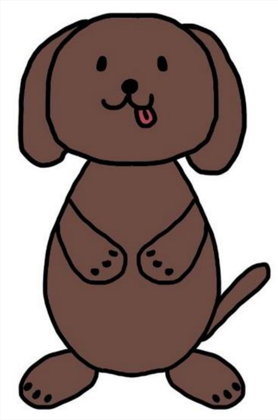
```
else{
  // change position variable
  position = "dog-cat";

  // change the images
  var leftImage = document.getElementById("left");
  leftImage.src = "dog.png";

  var rightImage = document.getElementById("right");
  rightImage.src = "cat.png";
}
```



Current position is cat-dog

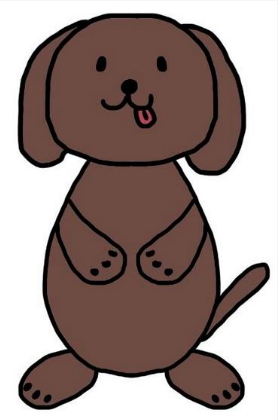
Cat & Dog 6

The web page displays a “Dog” picture.
When the user clicks the “Dog” picture, then it turns into a “Cat” picture.
If the user clicks the “Cat” picture, then it turns back to the “Dog” picture.



Cat & Dog 6

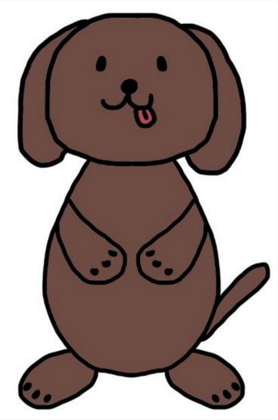
```
<img id="animal" src="dog.png" onClick="changeImage()" />
```



Cat & Dog 6

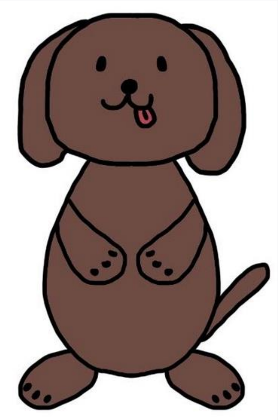
```
// variable to save the currently displayed animal
// two values: "dog" or "cat"
// original value is "dog"
var animal = "dog";
```

We use a **variable** to save the currently displayed animal



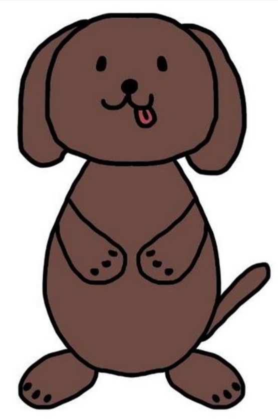
Cat & Dog 6

```
var animal = "dog";
function changeImage(){
  // check what is the current animal, then change it
  // change animal variable
  // change the image
}
```



Cat & Dog 6

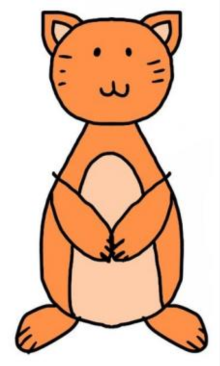
```
if(animal == "dog"){
  // change animal variable
  animal = "cat";

  // change the image
  var image = document.getElementById("animal");
  image.src = "cat.png";
}else...
```



Current animal is dog

Cat & Dog 6

```
else{
  // change animal variable
  animal = "dog";

  // change the image
  var image = document.getElementById("animal");
  image.src = "dog.png";
}
```



Current animal is cat