华中师范大学伍伦贡联合研究院
Central China Normal University Wollongong Joint Institute

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

CSIT884
Web Development

Lecture 03A - JavaScript Basics

Lecture slides are mainly taken from UOW

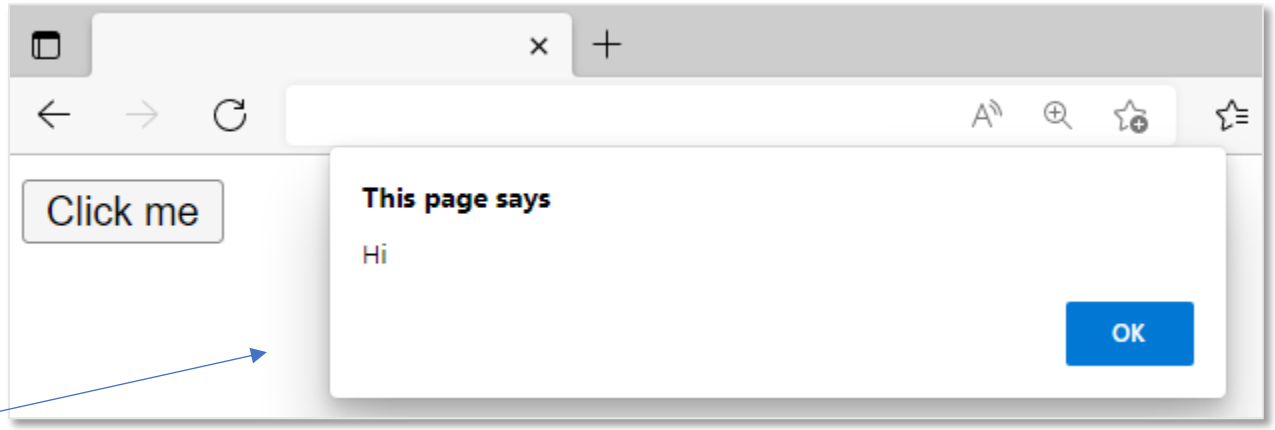
JavaScript

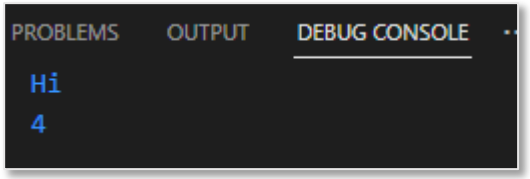
Objectives:

- Learn basic JavaScript programming language syntax
- Use JavaScript to make your website interactive

First JavaScript

```
<button onClick="sayHi();">
        Click me
</button>
<script>
    function sayHi(){
        alert("Hi");
        console.log("Hi");
        console.log(2+2);
    }
</script>
```





Where to include JavaScript

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');
    console.log(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 **s**`tudentName` and **S**`tudentName` 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 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

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");

}
```

```
var x = 5;

var y = 6;

if(x != y){

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

}else{

 alert("x and y are equal");

}
```

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");

}
```

- For-Loop statement:

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

 alert(i);

}
```

Useful tags for dynamic content:

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

Change content by 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:

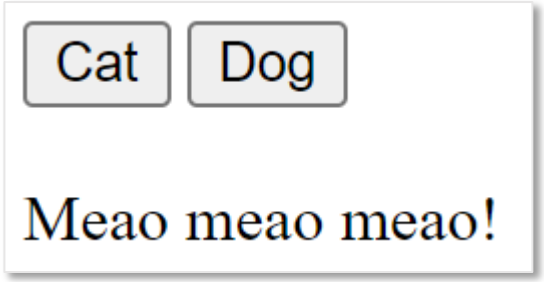
```
e.value = "the-new-value";
```

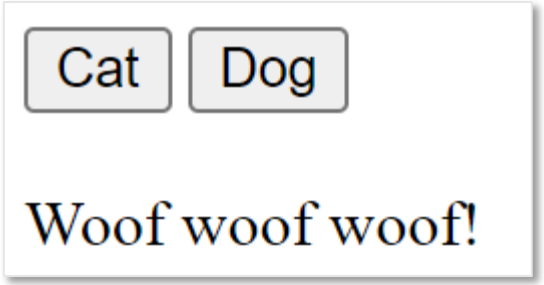
  for image:

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

Cat & Dog 1

- The web page displays **2 buttons**: "Cat" and "Dog".
  - If the user clicks the "Cat" button, a meao-meao message is displayed
  - If the user clicks the "Dog" button, a woof-woof message is displayed





Cat & Dog 1

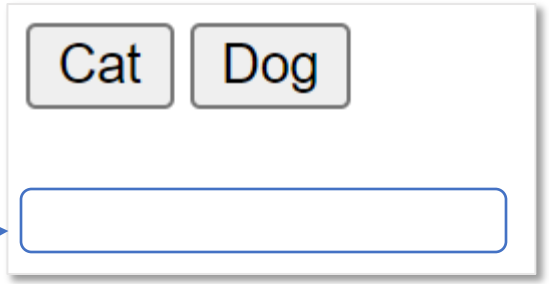
```
<button onclick="cat()">Cat</button>

<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

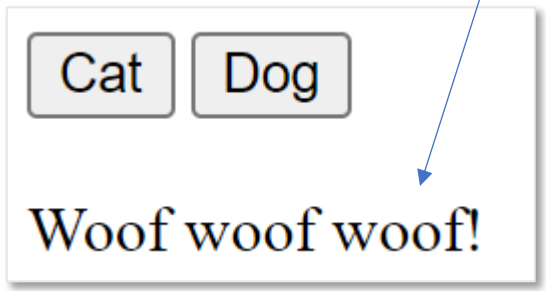
```
function dog(){
 // get the span element
 var displaySpan = document.getElementById("display");
 // show dog message
}
```




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

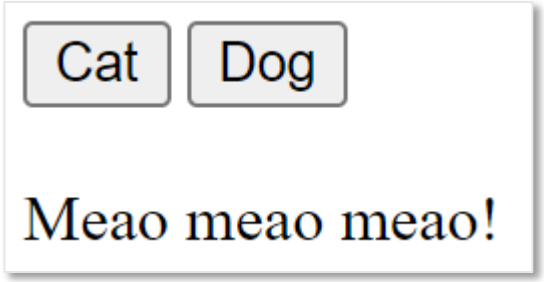
Cat & Dog 1

```
function dog(){

 // get the span element

 var displaySpan = document.getElementById("display");

 // show dog message

 displaySpan.innerHTML = "Woof woof woof!";

}
```



Cat & Dog 1

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


 // show cat message
 displaySpan.innerHTML = "Meao meao meao!";
}
```



Change content by 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:

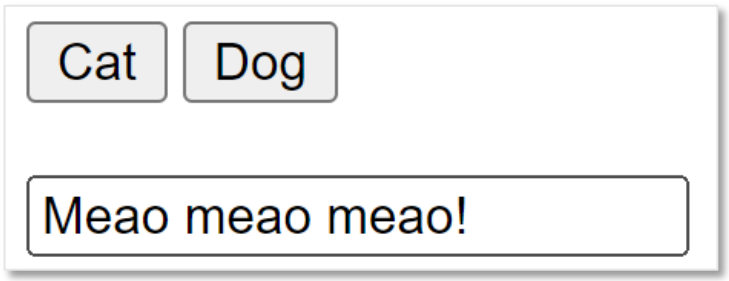
```
e.value = "the-new-value";
```

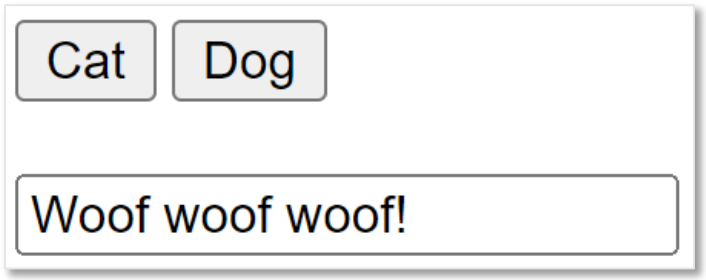
for image:

```
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 meao-meao message is displayed in the text field
  - If the user clicks the "Dog" button, a woof-woof message is displayed in the text field





Cat & Dog 2

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

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

<br /> <br />

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

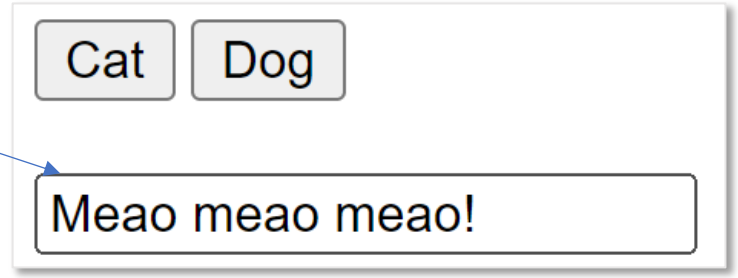
Cat & Dog 2

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


 // show cat message
}
```

Cat & Dog 2

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

 // show cat message
 displayField.value = "Meao meao meao!";
}
```

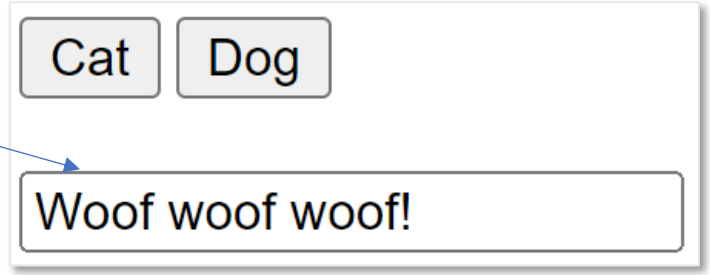
```
<input type="text" id="display" />
```



Cat & Dog 2

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

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

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



Change content by 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:

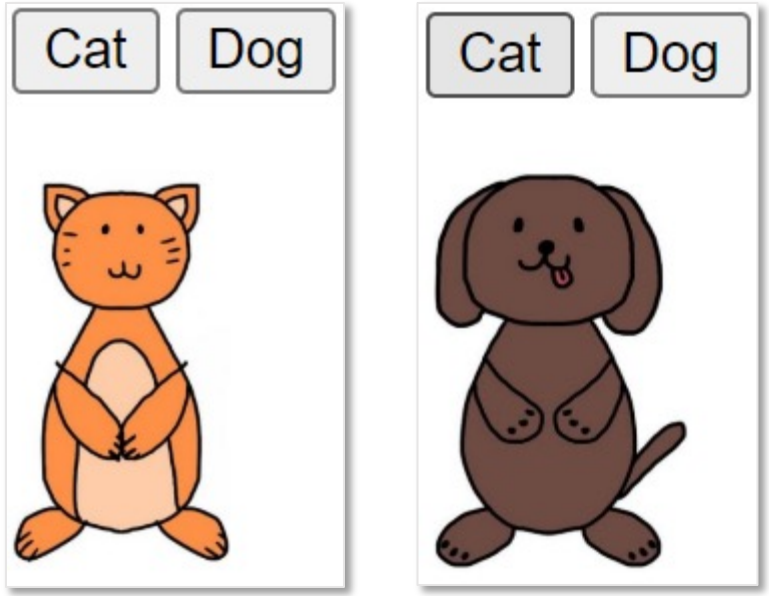
```
e.value = "the-new-value";
```

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, a cat picture is displayed
  - If the user clicks the "Dog" button, a dog picture is displayed



Cat & Dog 3

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

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

<br /> <br />

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

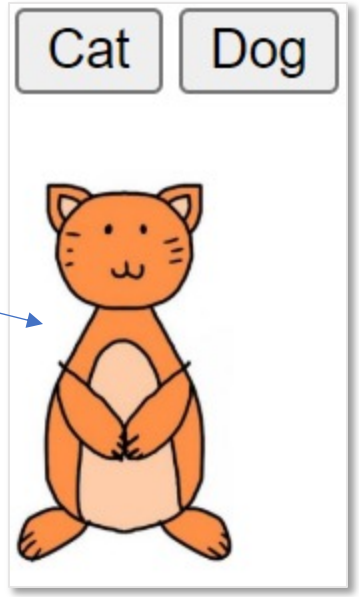
(empty image: no src)

Cat & Dog 3

```
function cat(){

 // get the image element


 // show cat picture

}
```

Cat & Dog 3

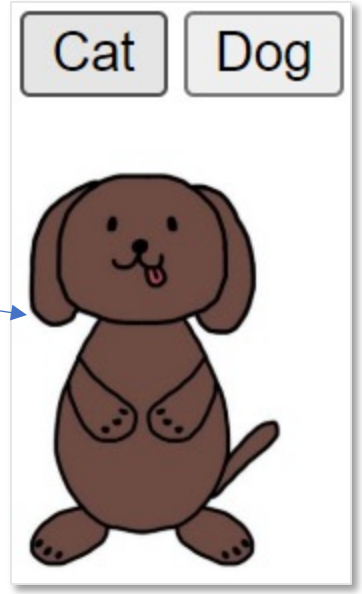
```
function cat(){
 // get the image element
 var image = document.getElementById("display");

 // 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" />
```

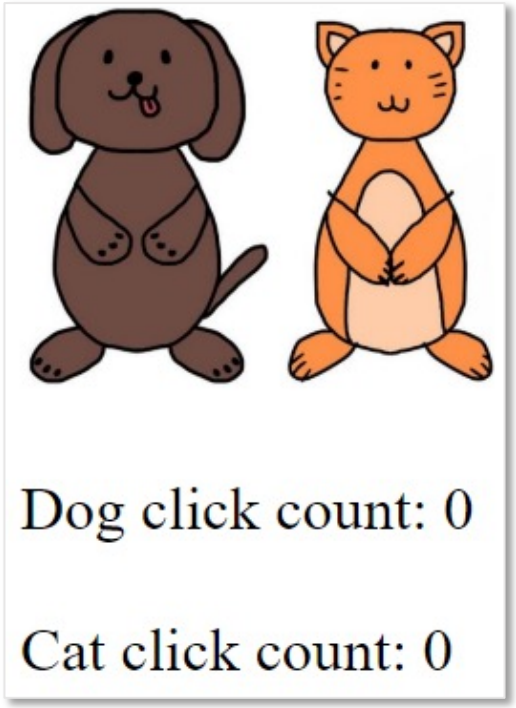
Using variables to save state information

- Sometime we use variables to save the current status of the page.

Cat & Dog 4

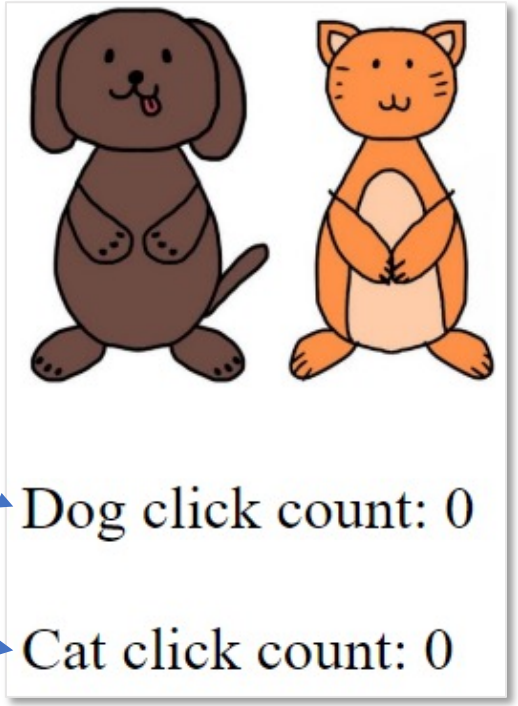
The web page displays **2 images**: " Cat " and "Dog", and **2 click counters**.

- 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.



Cat & Dog 4

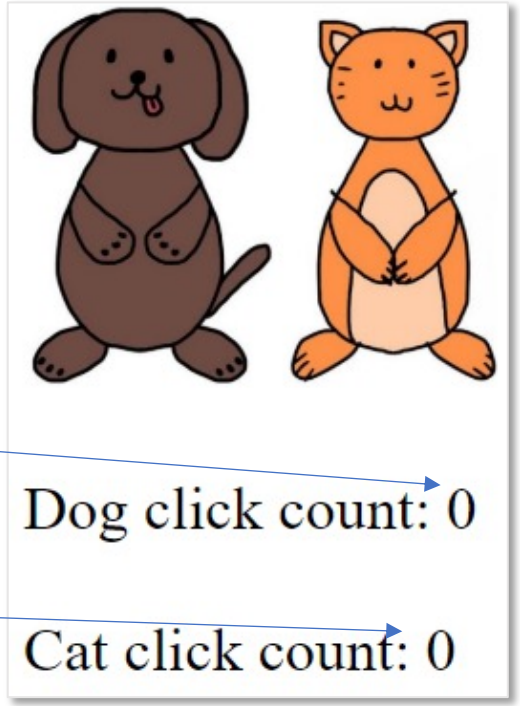
```
<img src="dog.png" onClick="dog()" />

<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>
```



Cat & Dog 4

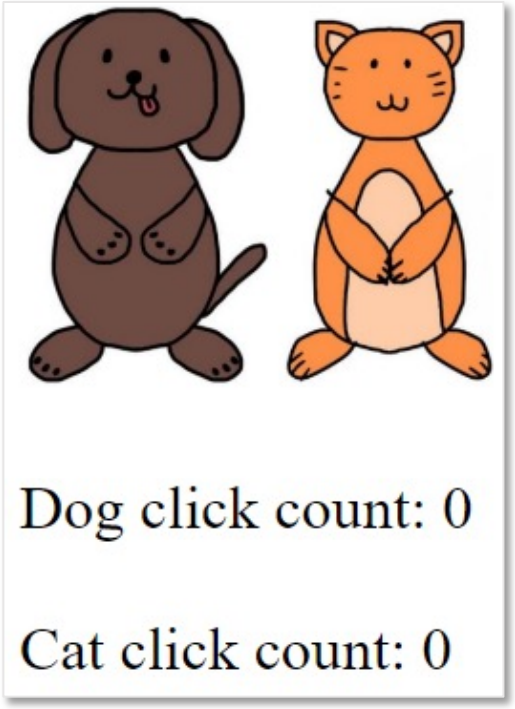
We use **variables** to save the current number of **dog-clicks** and **cat-clicks**.

```
// variable to save the number of dog clicks

var dogClick = 0;

// variable to save the number of cat clicks

var catClick = 0;
```



Cat & Dog 4

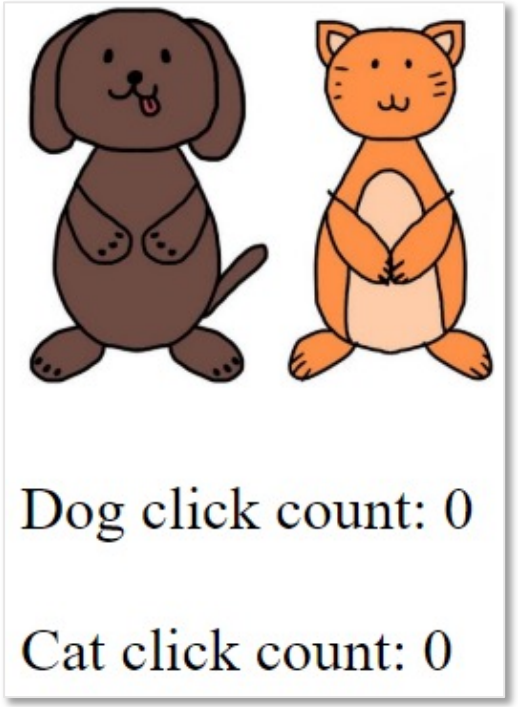
```
// variable to save the number of dog clicks

var dogClick = 0;

function dog(){

 // increase the number of dog clicks by 1

 // display the number of dog clicks

}
```



Cat & Dog 4

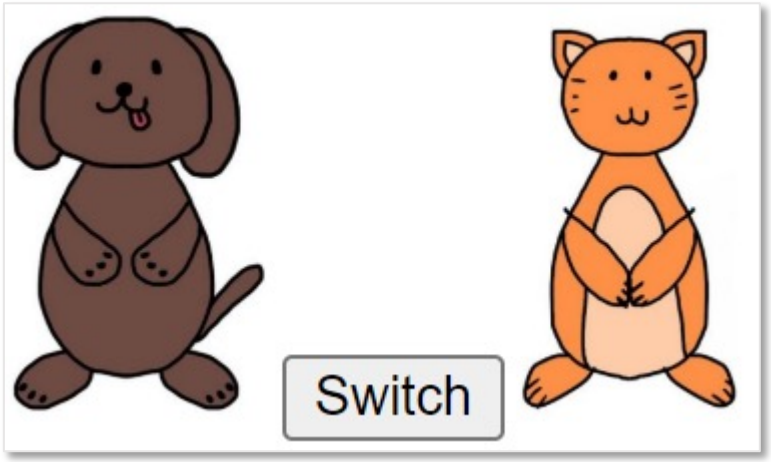
```
// variable to save the number of dog clicks

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;

}
```

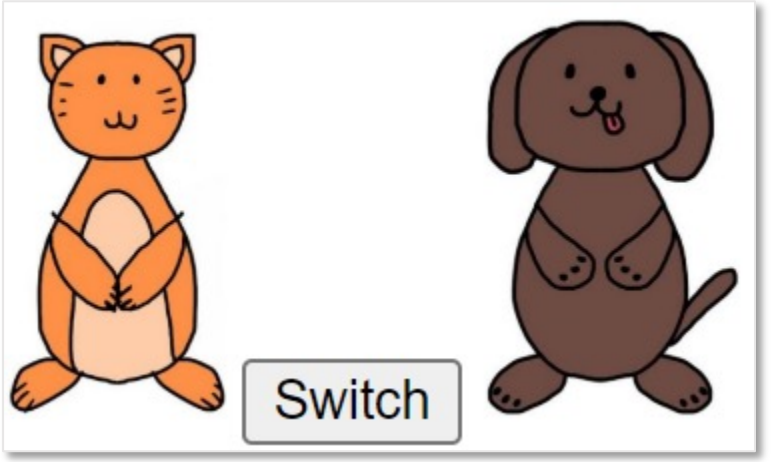
```
<span id="dogDisplay">0</span>
```



Cat & Dog 5

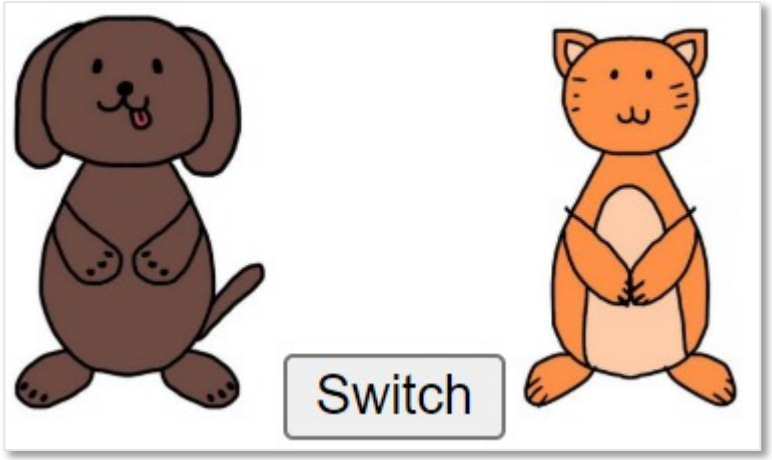
- The web page displays **2 images**: "Dog" on the left, "Cat" on the right, and **a button** "Switch".
- If the user clicks the "Switch" button, then the two images switch their places.





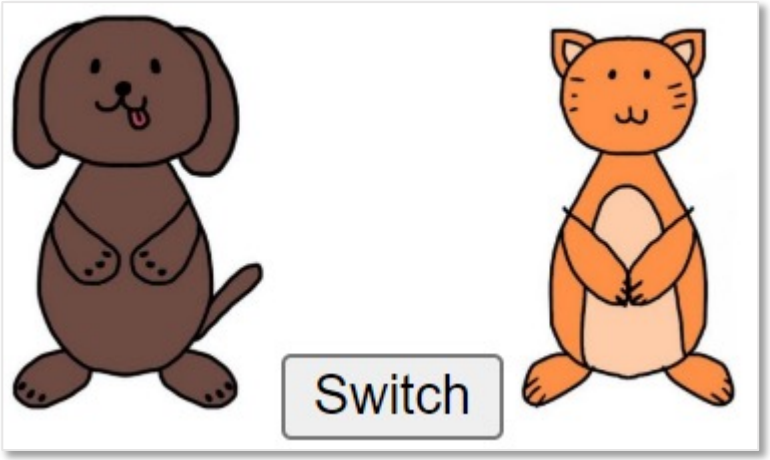
Cat & Dog 5

```
<img id="left" src="dog.png" />

<button onClick="switchImage()">

Switch

</button>

<img id="right" src="cat.png" />
```



Cat & Dog 5

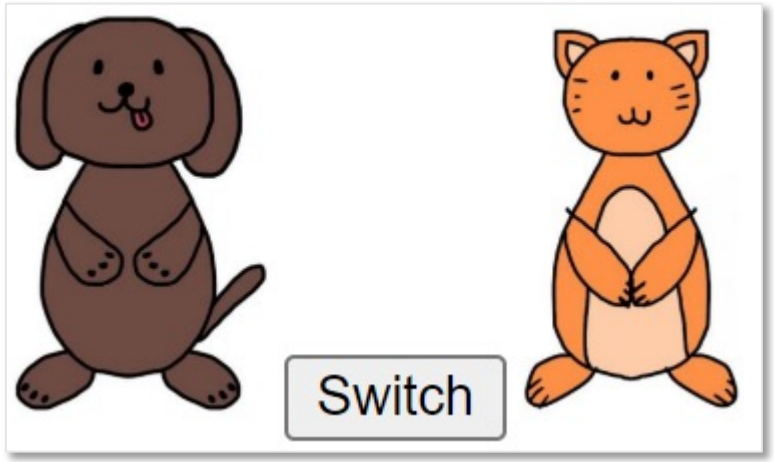
```
// variable to save the position of dog and cat images

// 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

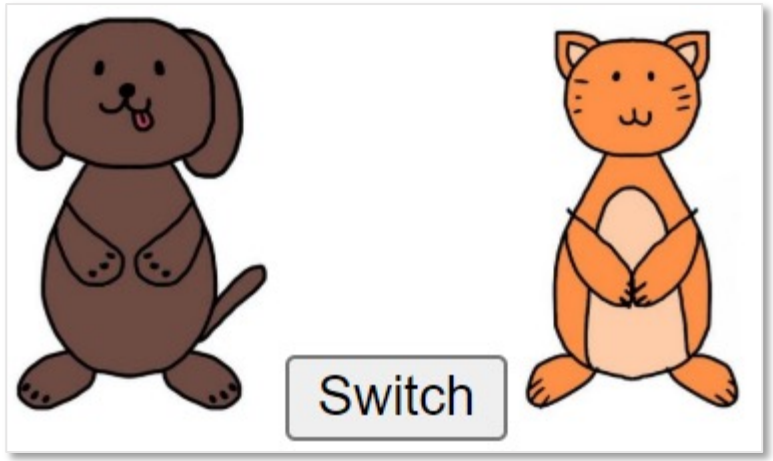
```
var position = "dog-cat";

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...
```

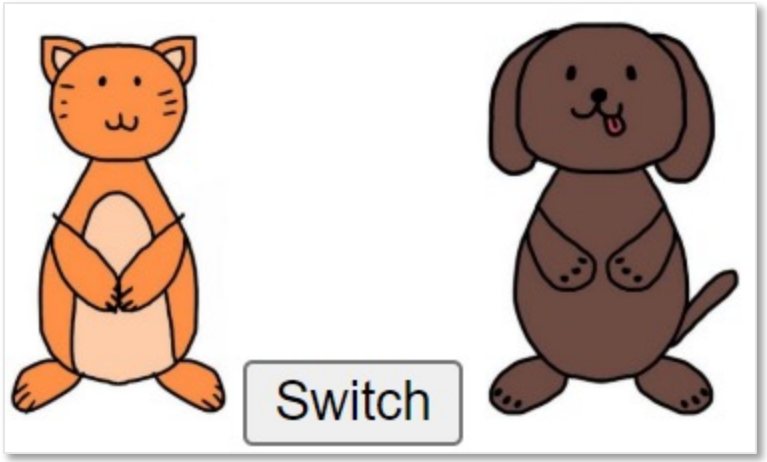
```
<img id="left" src="dog.png" />
<img id="right" src="cat.png" />
```



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";
}
```

```
<img id="left" src="dog.png" />
<img id="right" src="cat.png" />
```



Cat & Dog 6

The web page displays a "Dog" picture.

- If 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 current displayed animal

// two values: "dog" or "cat"

// original value is "dog"

var animal = "dog";
```

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

Cat & Dog 6

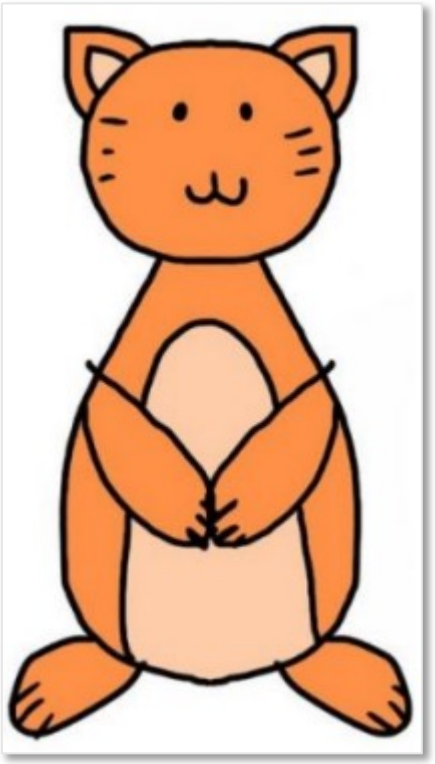
```
var animal = "dog";

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...
```

Cat & Dog 6

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



String

```
var text = "One Fish, Two Fish, Red Fish, Blue Fish";

var textLength = text.length;
                    → 39

var upper = text.toUpperCase();
                    → ONE FISH, TWO FISH, RED FISH, BLUE FISH

var lower = text.toLowerCase();
                     → one fish, two fish, red fish, blue fish

var fishIndex = text.indexOf("Fish");              → 4

var catIndex = text.indexOf("cat");                → -1

var redFound = text.includes("Red");               → true

var greenFound = text.includes("Green");           → false
```

String

```
var text = "One Fish, Two Fish, Red Fish, Blue Fish";

var s1 = text.slice(10, 12);   → Tw

var s2 = text.slice(10);       → Two Fish, Red Fish, Blue Fish

var s3 = text.slice(-9, -6);   → Blu

var s4 = text.slice(-9);       → Blue Fish
```

Date

There are several ways to create a `Date` object.

```
var d = new Date(); //current date & time

var d = new Date(millisec);

var d = new Date(dateString);

var d = new Date(year, month, day, hour, min, sec, millisec);
```

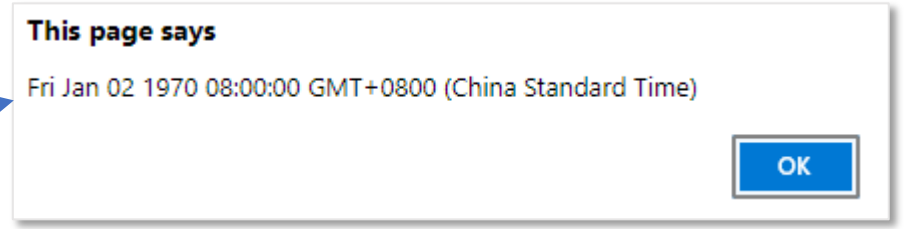
Date

```
var d = new Date(millisec);
```

Dates are calculated in milliseconds from 01 January, 1970 00:00:00 Universal Time (UTC).
One day contains 86,400,000 millisecond.

```
var d = new Date(86400000);
alert(d); //02 Jan 1970 00:00:00 UTC
```

(! Actual output depends on system and browser settings.)



Date

```
var d = new Date(dateString);


//using YYYY-MM-DD format

var d = new Date("2000-01-30");

alert(d);


//using YYYY-MM-DDTHH:MM:SS

var d = new Date("2000-01-30T10:00:00");

alert(d);
```

Date

```
var d = new Date(year, month, day, hour, min, sec, millisec);
```

The last 4 parameters can be omitted.

Months count from 0 to 11. January is 0. December is 11.

```
var d = new Date(2000, 0, 1); // 01 Jan 2000

alert(d);
```

Date

`getDate()`	Get the day as a number (1-31)
`getDay()`	Get the weekday as a number (0-6)
	Sunday is 0, Saturday is 6
`getFullYear()`	Get the four digit year (yyyy)
`getHours()`	Get the hours (0-23)
`getMilliseconds()`	Get the milliseconds (0-999)
`getMinutes()`	Get the minutes (0-59)
`getMonth()`	Get the month (0-11)
	January is 0, December is 11
`getSeconds()`	Get the seconds (0-59)
`getTime()`	Get the milliseconds since 01/Jan/1970

Date

```
var now = new Date();

alert("now is " + now);

alert("getDate returns " + now.getDate());

alert("getDay returns " + now.getDay());

alert("getFullYear returns " + now.getFullYear());

alert("getHours returns " + now.getHours());

alert("getMilliseconds returns " + now.getMilliseconds());

alert("getMinutes returns " + now.getMinutes());

alert("getMonth returns " + now.getMonth());

alert("getSeconds returns " + now.getSeconds());

alert("getTime returns " + now.getTime());
```

Date

`setDate()`	Set the day as a number (1-31)
`setFullYear()`	Set the year (optionally month and day)
`setHours()`	Set the hours (0-23)
`setMilliseconds()`	Set the milliseconds (0-999)
`setMinutes()`	Set the minutes (0-59)
`setMonth()`	Set the month (0-11)
`setSeconds()`	Set the seconds (0-59)
`setTime()`	Set the milliseconds since 01/Jan/1970

Date

```
var now = new Date();

alert(now);


var tomorrow = new Date();

tomorrow.setDate(now.getDate() + 1);

alert(tomorrow);


var hundredDaysAgo = new Date();

hundredDaysAgo.setDate(now.getDate() - 100);

alert(hundredDaysAgo);
```

Array

```
var arrayName = [item0, item1, ...];
```

```
var subjects = ["ISIT206", "MATH121", "CSCI301"];


subjects[1] = "LOGIC101"; //change the content of item 1
subjects[3] = "LAW201"; //add new item 3


alert(subjects[0]); //ISIT206
alert(subjects[1]); //LOGIC101
alert(subjects[2]); //CSCI301
alert(subjects[3]); //LAW201
```

Array

Length of array

```
var subjects = ["ISIT206", "MATH121", "CSCI301"];

// loop through an array

for(var i = 0; i < subjects.length; i++){

 alert(subjects[i]);

}
```

Array

```
var square = []; //empty array

for(var i = 0; i < 10; i++) {

 square[i] = i*i;

}

for(var i = 0; i < square.length; i++) {

 alert(square[i]);

}
```

Array

- **The `push()` method adds a new element to the end of an array**

```
var square = []; //empty array

for(var i = 0; i < 10; i++) {

 square.push(i*i);

}

for(var i = 0; i < square.length; i++) {

 alert(square[i]);

}
```

Array

```
var subjects = ["ISIT206", "MATH121", "CSCI301", "PHY211"];
```

- **The `indexOf(item)` method searches the array for the specified item, and returns its position**

```
var index = subjects.indexOf("MATH121");
```

- **The `splice(startIndex, howmany, item1, item2, itemN)` method changes the contents of an array by removing or replacing existing elements and/or adding new elements at a position**

```
var removedSubjects = subjects.splice(1, 2);

var removedSubjects = subjects.splice(1, 0, "CSIT884");
```

References

- http://www.w3schools.com/js
- https://developer.mozilla.org/en-US/docs/Web/JavaScript