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 05A - Dynamic Content Using JavaScript

Lecture slides are mainly taken from UOW

Objectives

- Use JavaScript to build website with dynamic content;
- Use JavaScript to generate animation on your website.

Recap: 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";
```

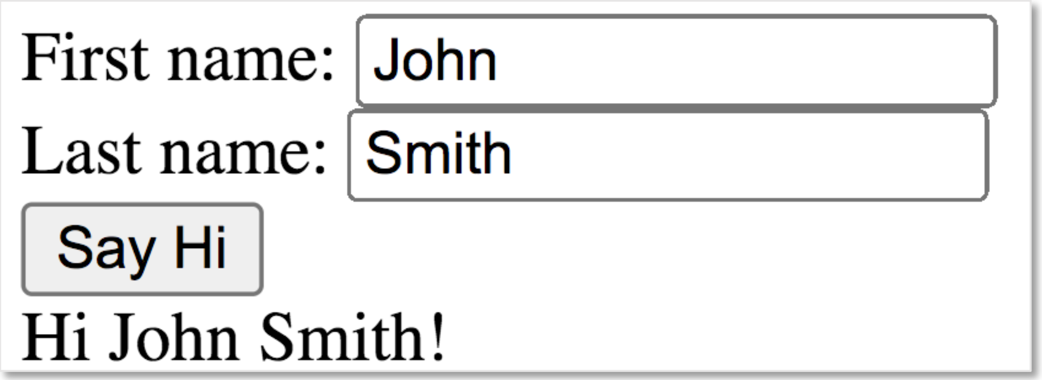
Useful tags for dynamic content:

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

Say Hi 1

The web page displays **2 text fields**: first name and last name, and **a button** *"Say Hi"*.

If the user enters *John* in the first name text field and *Smith* in the last name text field, and clicks the *"Say Hi"* button, then a greeting message is displayed on the page: *"Hi John Smith!"*.



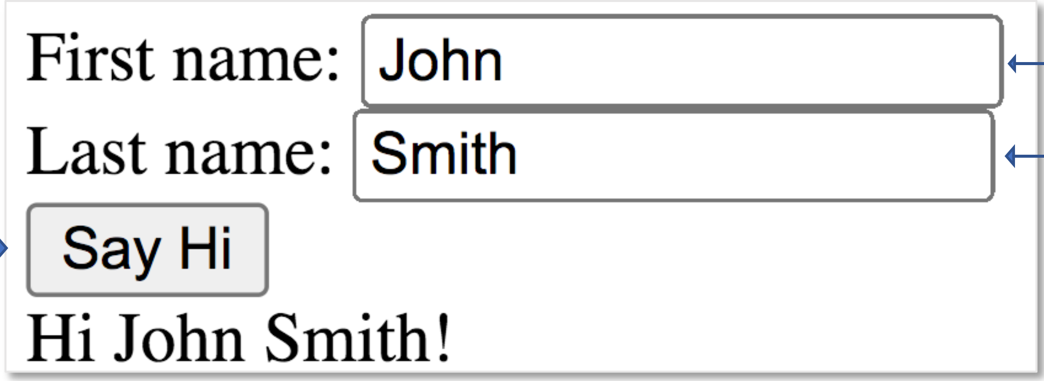
First name: John

Last name: Smith

Hi John Smith!

Say Hi 1

```
First name: <input type="text" id="firstname" /> <br />

Last name: <input type="text" id="lastname" /> <br />

<button onClick="sayHi()">Say Hi</button> <br />

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



First name: John

Last name: Smith

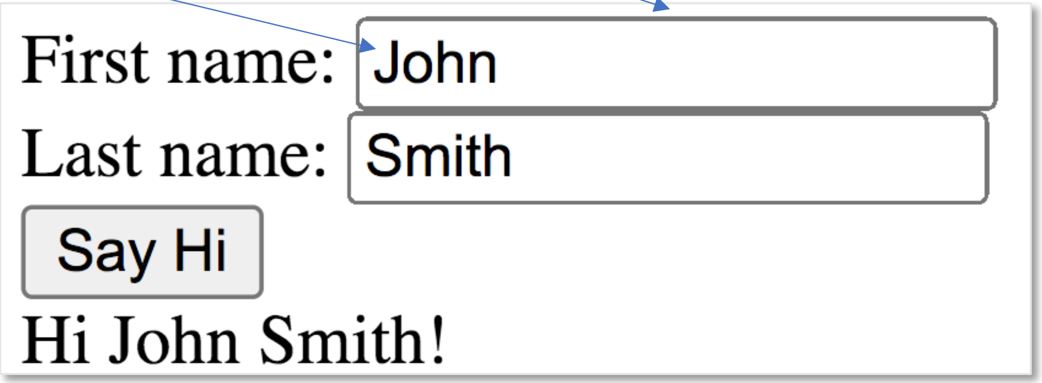
Hi John Smith!

Say Hi 1

```
function sayHi(){
 // get the first name


 // get the last name


 // construct the greeting message


 // display the greeting message
}
```

Say Hi 1

```
// get the first name

var firstNameInput = document.getElementById("firstname");

var firstName = firstNameInput.value;
```



First name: John

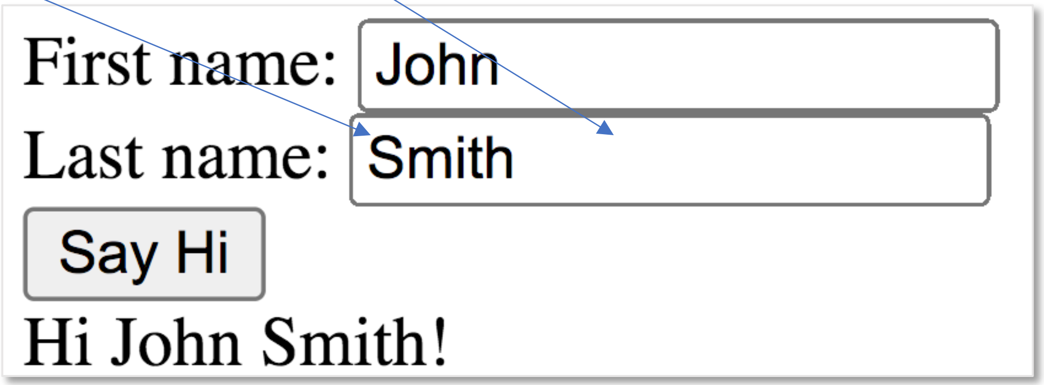
Last name: Smith

Hi John Smith!

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

Say Hi 1

```
// get the last name

var lastNameInput = document.getElementById(”lastname");

var lastName = lastNameInput.value;
```



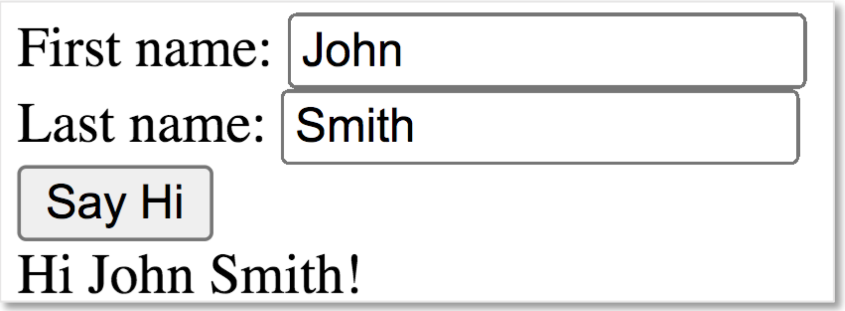
First name: John

Last name: Smith

Hi John Smith!

```
<input type="text" id=”lastname" />
```

Say Hi 1

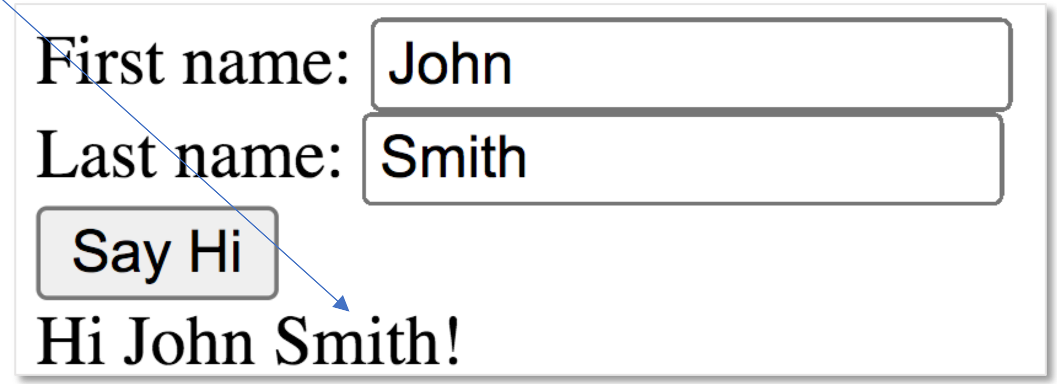


First name: John
Last name: Smith
Say Hi
Hi John Smith!

```
// get the first name
var firstNameInput = document.getElementById("firstname");
var firstName = firstNameInput.value;

// get the last name
var lastNameInput = document.getElementById("lastname");
var lastName = lastNameInput.value;

// construct the greeting message
var greetingMessage = "Hi " + firstName + " " + lastName + "!";
```

Say Hi 1

```
// construct the greeting message
var greetingMessage = "Hi " + firstName + " " + lastName + "!";


// display the greeting message
var greetingSpan = document.getElementById("greeting");
greetingSpan.innerHTML = greetingMessage;
```



First name: John

Last name: Smith

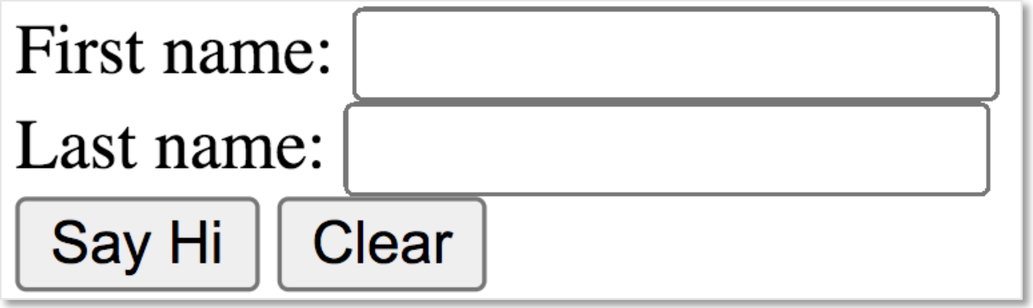
Hi John Smith!

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

Say Hi 2

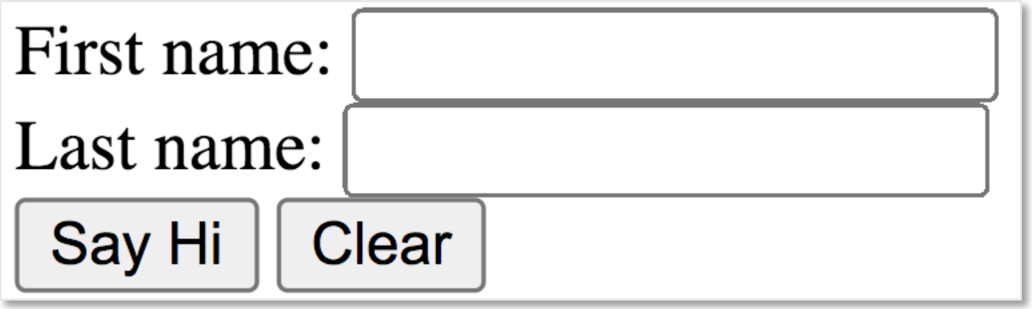
Add a button “Clear”.

If the user clicks the “Clear” button then the texts in the input fields and the greeting message are removed.



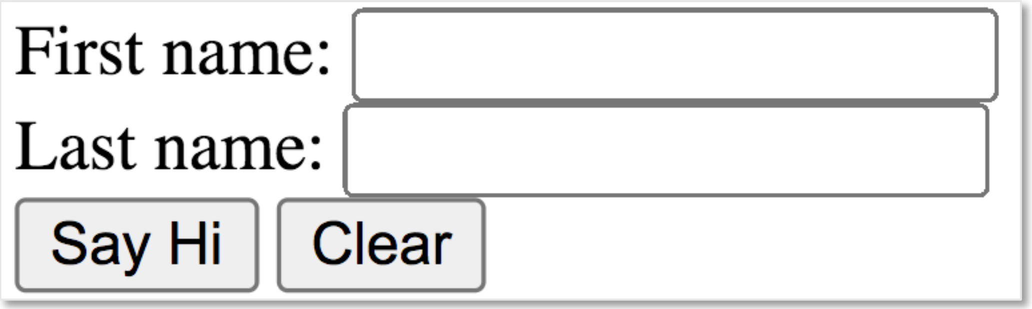
Say Hi 2

```
First name: <input type="text" id="firstname" /> <br /><br />

Last name: <input type="text" id="lastname" /> <br /><br />

<button onClick="sayHi()">Say Hi</button>

<button onClick="clearPage()">Clear</button> <br /><br />

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



Say Hi 2

```
function clearPage(){
 // clear the firstname text field


 // clear the lastname text field


 // clear the greeting message
}
```

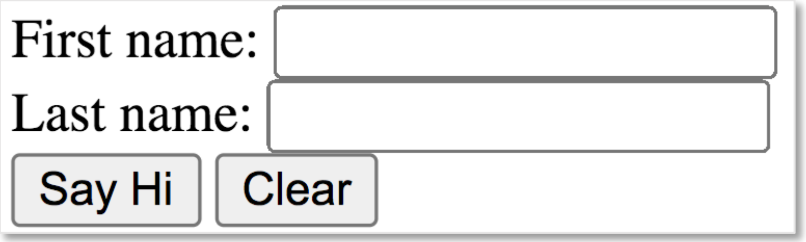


First name:

Last name:

Say Hi 2

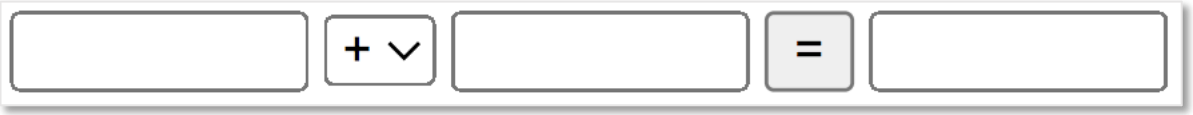
```
// clear the firstname text field
var firstNameInput = document.getElementById("firstname");
firstNameInput.value = "";

// clear the lastname text field
var lastNameInput = document.getElementById("lastname");
lastNameInput.value = "";

// clear the greeting message
var greetingSpan = document.getElementById("greeting");
greetingSpan.innerHTML = "";
```



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

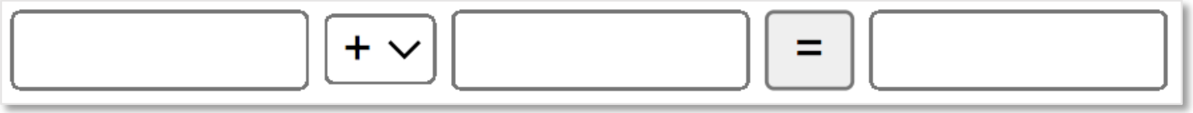
Math Question

- The user enters 2 numbers into the two text fields,
- then selects the operation (+, - , x) from the drop-down list,
- then clicks the “=” button,
- then answer will be generated.



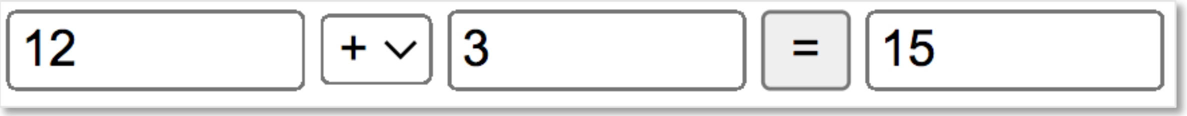
Math Question

```
<input type="text" id="input1" />
<select id="operationSelect">
 <option value="add">+</option>
 <option value="subtract">-</option>
 <option value="multiply">x</option>
</select>
<input type="text" id="input2" />
<button onClick="answer()">
=
</button>
<input type="text" id="result" />
```



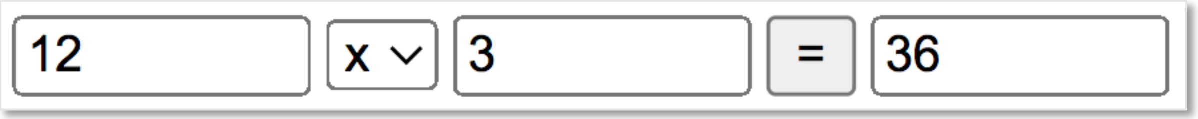
Math Question

```
function answer(){
  // get the 1st number
  // get the 2nd number
  // get the operation
  // calculate the result
  // display the result
}
```



12 + 3 = 15

Math Question

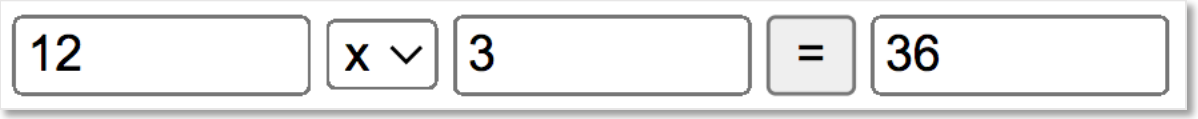


12 x 3 = 36

```
// get the 1st number
var inputField1 = document.getElementById("input1");
var number1 = Number(inputField1.value);

// get the 2nd number
var inputField2 = document.getElementById("input2");
var number2 = Number(inputField2.value);

// get the operation
var operationSelect = document.getElementById("operationSelect");
var operation = operationSelect.value;
```

Math Question

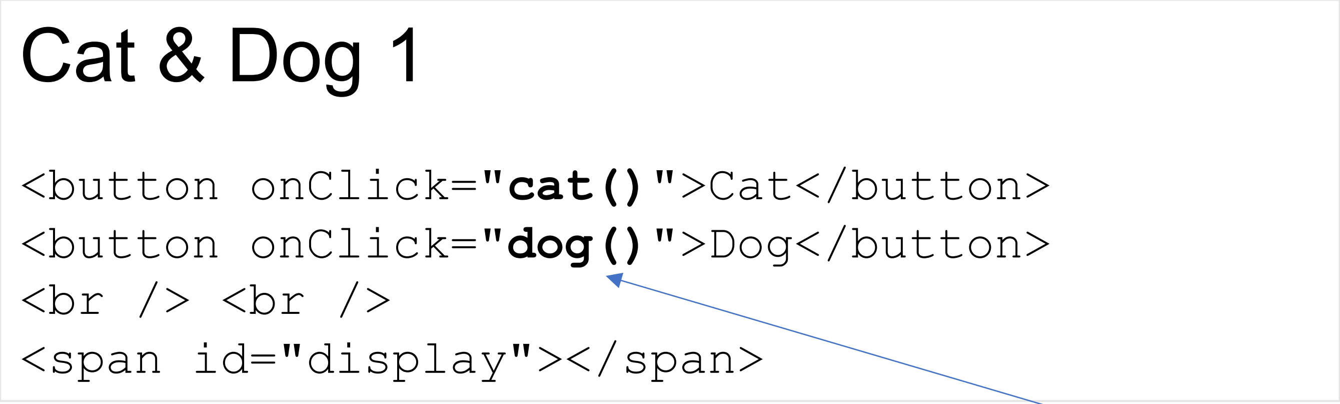
```
// calculate the result
var result;
if(operation == "add"){
 result = number1 + number2;
}else if(operation == "subtract"){
 result = number1 - number2;
}else if(operation == "multiply"){
 result = number1 * number2;
}

// display the result
var resultField = document.getElementById("result");
resultField.value = result;
```



12 x 3 = 36

Passing parameters to function

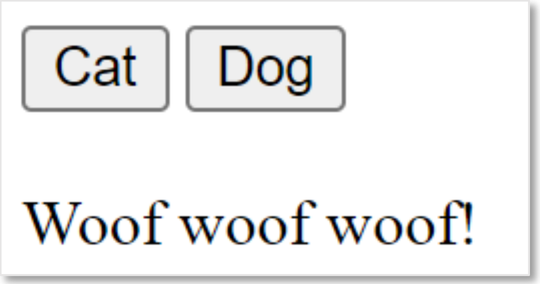
Sometimes, it is useful to pass **parameters** to the **function**



Cat & Dog 1

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

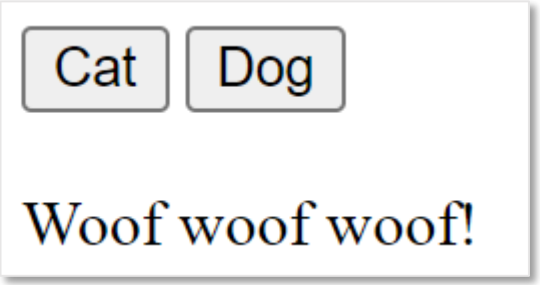
Old solution: using two functions



Woof woof woof!

Cat & Dog 1B

```
<button onClick="showMessage('Meao meao meao!')">Cat</button>

<button onClick="showMessage('Woof woof woof!')">Dog</button>

<br /> <br />

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



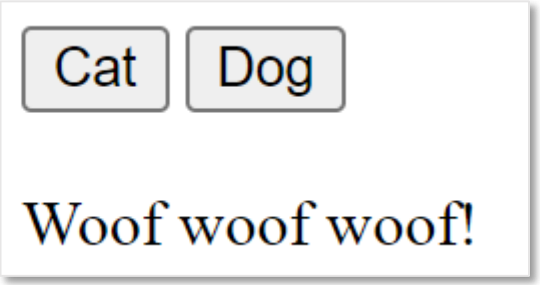
Woof woof woof!

New solution: using one function with parameter

Cat & Dog 1B

New solution: using one function with parameter

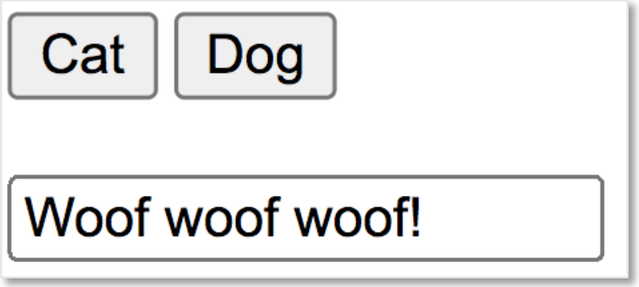
```
function showMessage(message){
 // get the span element
 var displaySpan = document.getElementById("display");
 // show the message
 displaySpan.innerHTML = message;
}
```



Woof woof woof!

Cat & Dog 2B

```
<button onClick="showMessage('Meao meao meao!')">Cat</button>
<button onClick="showMessage('Woof woof woof!')">Dog</button>
<br /> <br />
<input type="text" id="display" />
```

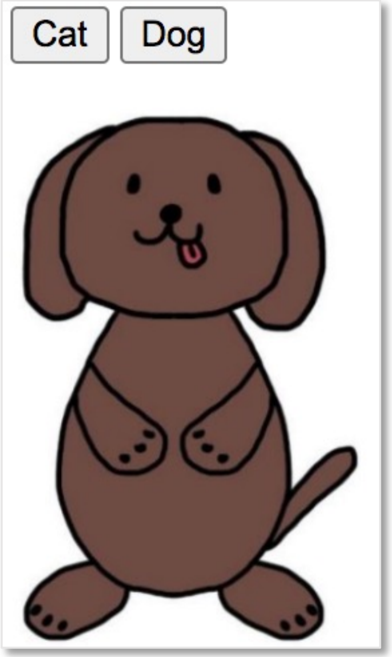


Woof woof woof!

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

 // show the message
 displayField.value = message;
}
```

Cat & Dog 3B

```
<button onClick="showImage('cat.png')">Cat</button>

<button onClick="showImage('dog.png')">Dog</button>

<br /> <br />

<img id="display" />



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

 // show the animal picture
 image.src = imageFile;
}
```



Random

- To get a random (decimal) number from 0 (inclusive) to 1 (exclusive):

```
var x = Math.random(); // random decimal number in [0,1)
```

Random

- Random number in the range`[0,1)`:

```
      Math.random()
```

- Random number in the range`[0,10)`:

```
      Math.random()*10
```

- Random **integer** in the range`[0,10)`:

```
      Math.floor(Math.random()*10)
```

```
//generate a random integer: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
var x = Math.floor(Math.random() * 10);
```

Random

- Random number in the range`[0,1)`:

```
Math.random()
```

- Random number in the range`[0,10)`:

```
Math.random()*10
```

- Random **integer** in the range`[0,10)`:

```
Math.floor(Math.random()*10)
```

- Random **integer** in the range`[1,11)`:

```
Math.floor(Math.random()*10)+1
```

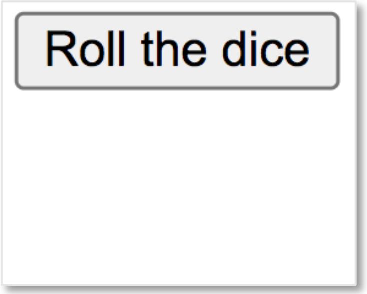
```
//generate a random integer: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
var x = Math.floor(Math.random() * 10) + 1;
```

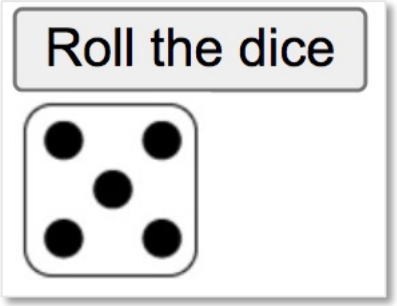
Dice 1

- When the button is clicked, a random dice is displayed.

Generate a random integer from 1 to 6:

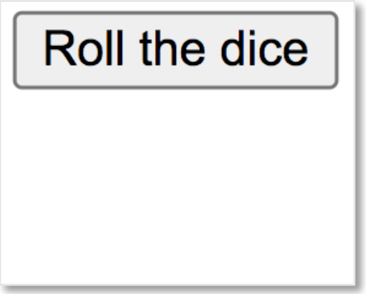
```
Math.floor(Math.random() * 6) + 1
```

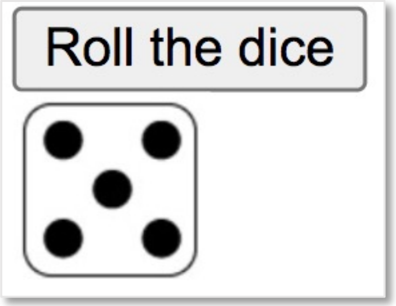




Dice 1

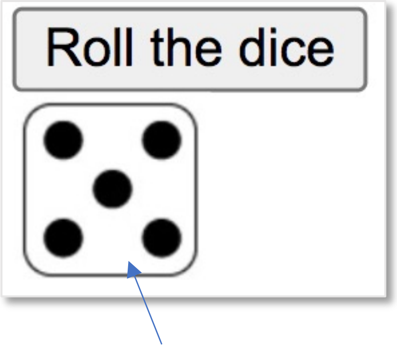
```
<button onClick="rollDice()">

Roll the dice

</button>

<br />

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





Dice 1

```
function rollDice(){
 // generate a random dice value from 1 to 6
 var diceValue = Math.floor(Math.random() * 6) + 1;

 // get image file name for this dice value
 var imageFile = "dice" + diceValue + ".png";

 // show the image
 var diceImage = document.getElementById("dice");
 diceImage.src = imageFile;
}
```



dice5.png

Animation

Start an animation

```
var animationSchedule = setInterval(animationFunction, milisecs);
```

- The `setInterval()` method repeatedly calls a function or executes a code snippet, with a fixed time delay between each call

We need to specify 2 things:

- ***what needs to be done***: write a function to do the task
- ***how often***: how often this task needs to be done

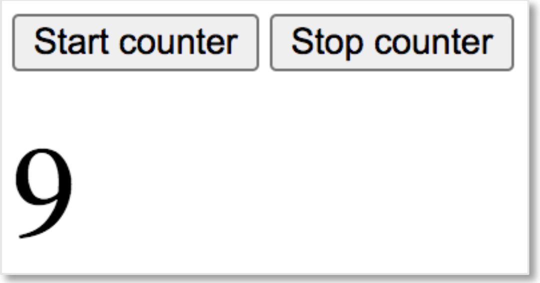
Stop the animation

```
clearInterval(animationSchedule);
```

Counter animation

- When the “Start Counter” button is clicked, the counter animation displays a number increasing every 1 second.
- When the “Stop Counter” button is clicked, the number stops increasing.

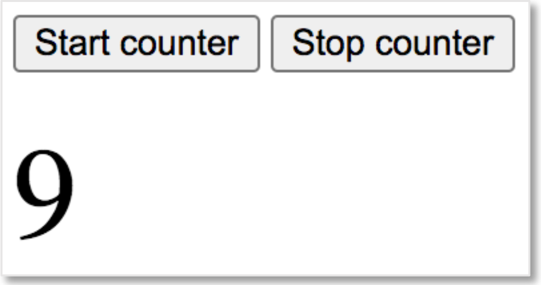
```
initially counter = 0
start animation
1000 milisec              showCounter()    counter = 1
1000 milisec              showCounter()    counter = 2
1000 milisec              showCounter()    counter = 3
…
stop animation
```



9

Counter animation

```
<button onClick="startCounterAnimation()">

Start counter

</button>

<button onClick="stopCounterAnimation()">

Stop counter

</button>

<br /><br />

<font size="7">

<span id="counter"></span>

</font>
```

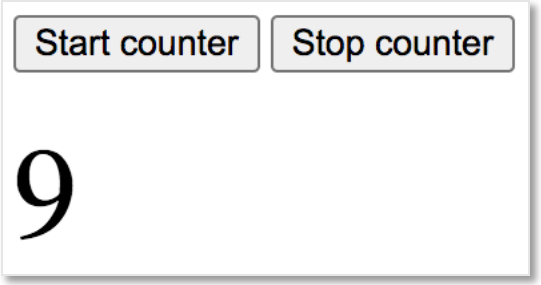


Counter animation

```
var counter = 0;
var counterSchedule;
function startCounterAnimation(){
 // start the counter animation
 counterSchedule = setInterval(showCounter, 1000);
}
```

Start an animation

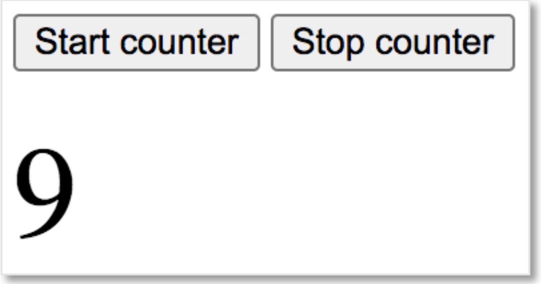
- ***what needs to be done:*** write a function to do the task
- ***how often:*** how often this task need to be done



Counter animation

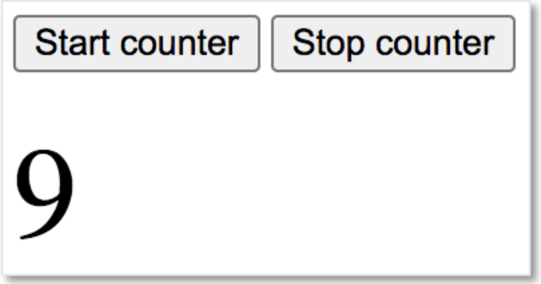
what needs to be done:
write a function to do the task of the animation

```
function showCounter(){
 // increase the counter by 1
 counter = counter + 1;
 // show the counter
 var counterSpan = document.getElementById("counter");
 counterSpan.innerHTML = counter;
}
```



Counter animation

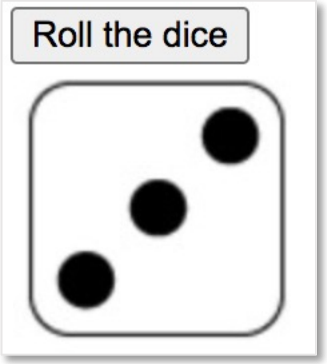
```
function stopCounterAnimation(){
 clearInterval(counterSchedule);
}
```



Dice 2 - animation

- When the button is clicked, within 1 second, a flash of 10 random dice images are displayed, and then it stops.

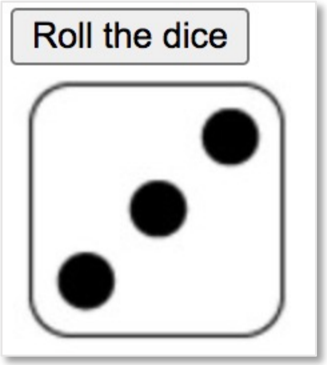
```
start animation                         counter = 0
    100 milisec          rollDice()     counter = 1
    100 milisec          rollDice()     counter = 2
    100 milisec          rollDice()     counter = 3
    100 milisec          rollDice()     counter = 4
    100 milisec          rollDice()     counter = 5
    100 milisec          rollDice()     counter = 6
    100 milisec          rollDice()     counter = 7
    100 milisec          rollDice()     counter = 8
    100 milisec          rollDice()     counter = 9
    100 milisec          rollDice()     counter = 10

stop animation
```



Use a **counter variable** to know when to stop the animation

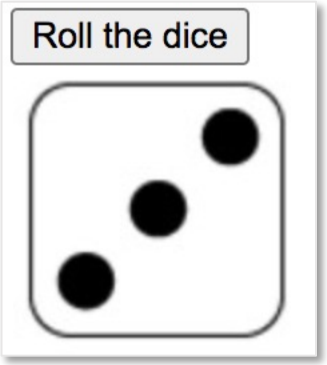
Dice 2 - animation

```
<button onClick="rollDiceAnimation()">

Roll the dice

</button>

<br />

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



Dice 2 - animation

```
var rollDiceSchedule;
var rollDiceCounter;
function rollDiceAnimation(){
 // set the roll dice counter to 0
 rollDiceCounter = 0;

 // start the roll dice animation
 rollDiceSchedule = setInterval(rollDice, 100);
}
```



Start an animation

what needs to be done: write a function to do the task

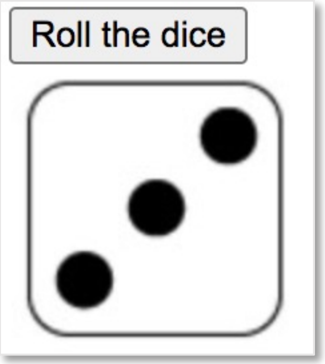
how often: how often this task need to be done

Dice 2 - animation

what needs to be done:

write a function to do the task of the animation

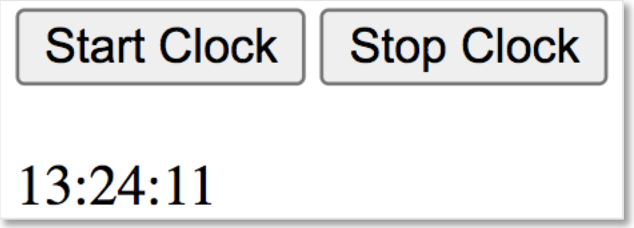
```
function rollDice(){
 // generate a random dice value from 1 to 6
 var diceValue = Math.floor(Math.random() * 6) + 1;

 // get image file name for this dice value
 var imageFile = "dice" + diceValue + ".png";

 // show the image
 var diceImage = document.getElementById("dice");
 diceImage.src = imageFile;

 // increase the roll dice counter
 rollDiceCounter = rollDiceCounter + 1;

 // if the roll dice counter reaches 10 then stop the animation
 if(rollDiceCounter == 10){
  clearInterval(rollDiceSchedule);
 }
}
```



Digital Clock - animation

When the “Start Clock” button is clicked, the clock animation displays the current time every 1 second. When the “Stop Clock” button is clicked, the clock stops.

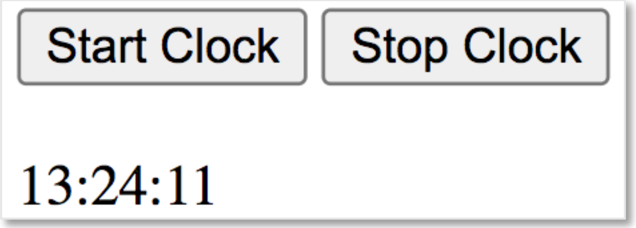
```
start animation
    1000 milisec           showClock()
    1000 milisec           showClock()
    1000 milisec           showClock()
    1000 milisec           showClock()
    1000 milisec           showClock()
    …
stop animation
```



13:24:11

Digital Clock - animation

```
<button onClick="startClockAnimation()">

Start Clock

</button>

<button onClick="stopClockAnimation()">

Stop Clock

</button>

<br /><br />

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



13:24:11

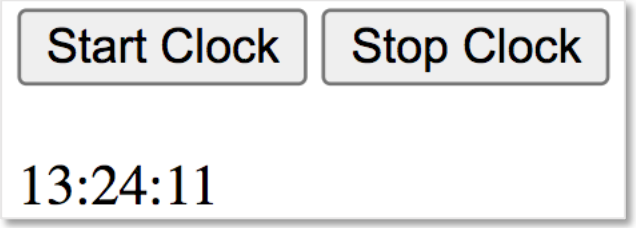
Digital Clock - animation

```
var clockAnimation;

function startClockAnimation(){

  // start the clock animation

  clockAnimation = setInterval(showClock, 1000);

}
```

Start an animation

what needs to be done: write a function to do the task

how often: how often this task need to be done



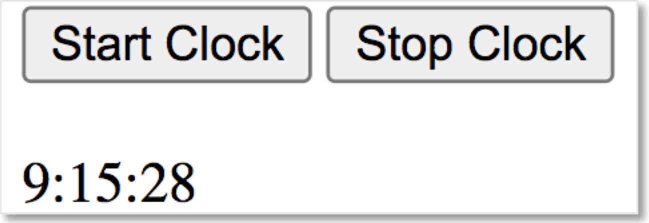
13:24:11

Digital Clock - animation

```
function showClock(){
// get current date and time
var now = new Date();
// formulate the clock Hour:Minute:Second
var clock = now.getHours() + ":" + now.getMinutes() + ":" + now.getSeconds();
// show the clock
var clockSpan = document.getElementById("clock");
clockSpan.innerHTML = clock;
}
```

what needs to be done:

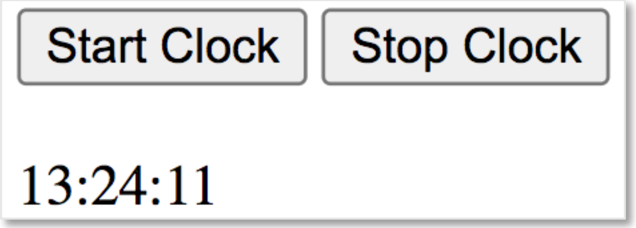
write a function to do the task



```
9:15:28
```

Digital Clock - animation

```
function stopClockAnimation(){
 clearInterval(clockAnimation);
}
```

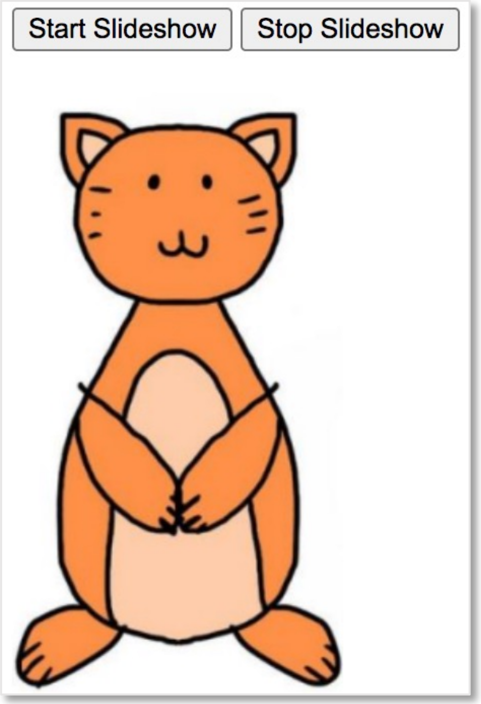


13:24:11

Slideshow - animation

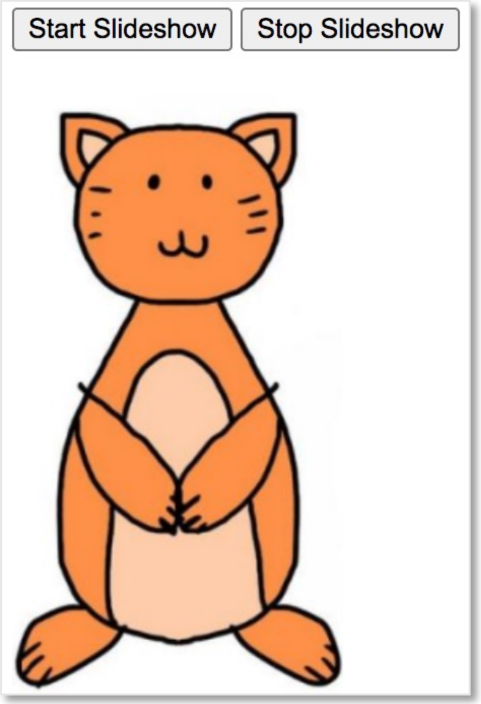
When the “Start Slideshow” button is clicked, random images will be displayed every 1 second. When the “Stop Slideshow” button is clicked, the slide show stops.

```
start animation
    1000 milisec          showImage()
    1000 milisec          showImage()
    1000 milisec          showImage()
    1000 milisec          showImage()
    1000 milisec          showImage()
    …
stop animation
```

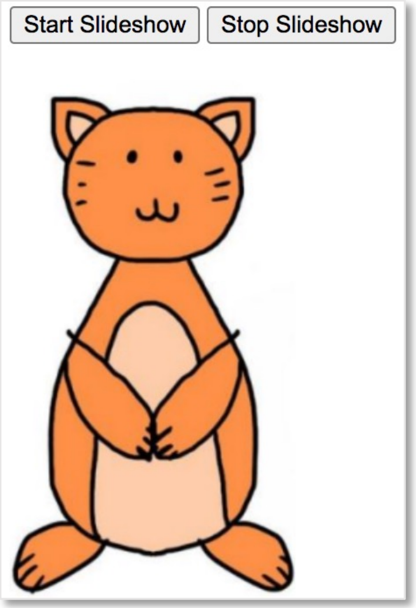


Slideshow - animation

```
<button onClick="startAnimation()">
Start Slideshow
</button>
<button onClick="stopAnimation()">
Stop Slideshow
</button>
<br /><br />
<img id="imageHolder" height="500px" />
```



Slideshow - animation



```
var slideshowAnimation;

function startAnimation(){

  // start the slideshow animation

  slideshowAnimation = setInterval(showImage, 1000);

}
```

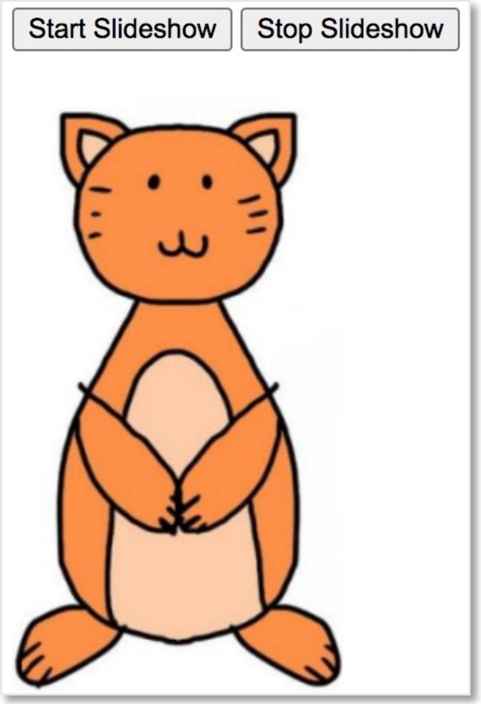
Start an animation

what needs to be done: write a function to do the task

how often: how often this task need to be done

Slideshow - animation

```
// list of images for the slideshow animation
var IMAGE_LIST = [
 "cat.png",
 "dog.png",
 "heart.png",
 "help.png",
 "home.png"
];
```

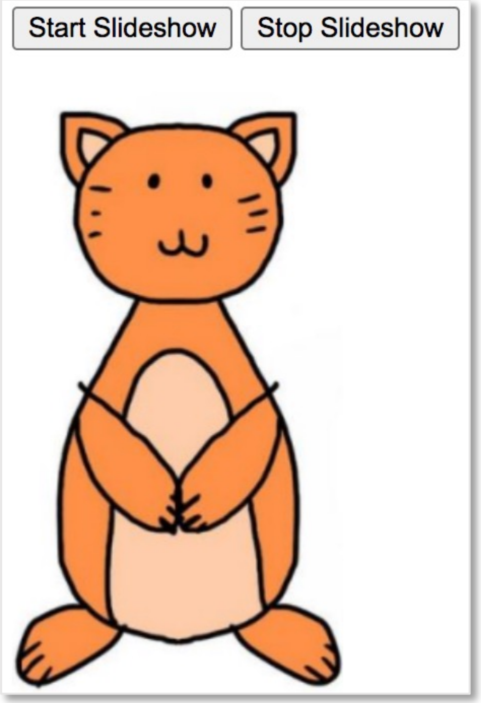


Slideshow - animation

```
// list of images for the slideshow animation
var IMAGE_LIST = [
 …
];


function showImage(){
 // get a random image index
 var index = Math.floor(Math.random() * IMAGE_LIST.length);
 // show the image
 var imgElement = document.getElementById("imageHolder");
 imgElement.src = IMAGE_LIST[index];
}
```

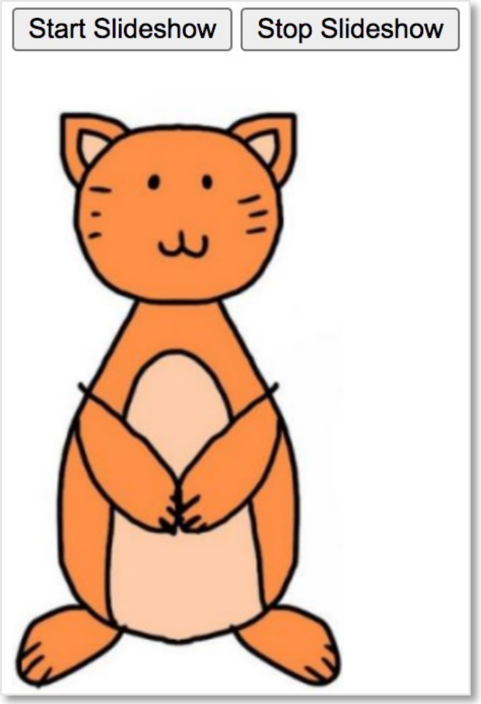
what needs to be done:

write a function to do the task



Slideshow - animation

```
function stopAnimation(){

 clearInterval(slideshowAnimation);

}
```



References

- Robert W. Sebesta, Programming the World Wide Web, 8th edition, Pearson, 2015.
- Jennifer Niederst Robbins, Learning Web Design - A Beginner's guide to HTML, CSS, JavaScript and Web Graphics, 5th edition, O’Reilly Media, 2018.
- http://www.w3schools.com/js
- http://developer.mozilla.org/en-US/docs/Web/JavaScript