



Week 10

**DECO 1400/7140 Intro  
to Web Design**

University of Queensland

---

Semester 1, 2022



# JavaScript Click & Form Events

# Programming Fundamentals in JavaScript: Looping



Repeating Code

Repeating Code

Repeating Code

Repeating Code

Repeating Code

Repeating Code

Repeating Code

Repeating Code

Repeating Code

Repeats until a specified condition evaluates to false:

```
let a = 0;  
for(let i = 0; i < 5; i++) {  
    a = a + 3;  
}  
console.log(a);
```

The diagram illustrates the three components of a for loop using yellow curved arrows:

- Initial Expression:** Points to the `let i = 0;` part of the for loop.
- Condition:** Points to the `i < 5;` part of the for loop.
- Increment:** Points to the `i++` part of the for loop.

# For Loops

```
let a = 0;
for(let i = 0; i < 5; i++) {
  a = a + 3;
}
console.log(a);
```


LOOP	VALUE OF i	i < 5	VALUE OF a
-	-	-	0
1	0	T	3
2	1	T	6
3	2	T	9
4	3	T	12
5	4	T	15
6	5	F	-

LOOP	VALUE OF i	i < 4	VALUE OF total
-	-	-	0
1	0	T	34
2	1	T	77
3	2	T	105
4	3	T	128
5	4	F	-

```
const marks = [34, 43, 28, 23];
let total = 0;
for(let i = 0; i < marks.length; i++) {
    total = total + marks[i];
}
let average = total / marks.length;
```

Iterate while condition is true:

```
let a = 2;  
while(a < 19) {  
    a = a + 3;  
}
```



Condition

A yellow curved arrow originates from the condition `a < 19` in the `while` statement and points to the word "Condition" on the right side of the slide.

# While Loops



```
let a = 2;  
while(a < 19) {  
    a = a + 3;  
}
```

LOOP	a < 19	VALUE OF a
-	-	2
1	T	5
2	T	8
3	T	11
4	T	14
5	T	17
6	T	20
7	F	-



<https://jsfiddle.net/mattluscombe/6fwygsen>

Simpler alternative  
for iterating over  
a set length of values/elements

- Creates variable **for** each iteration
- Separates variable & iterable  
with **of** statement

## For Of/Each Loops

# Beware of the Endless!

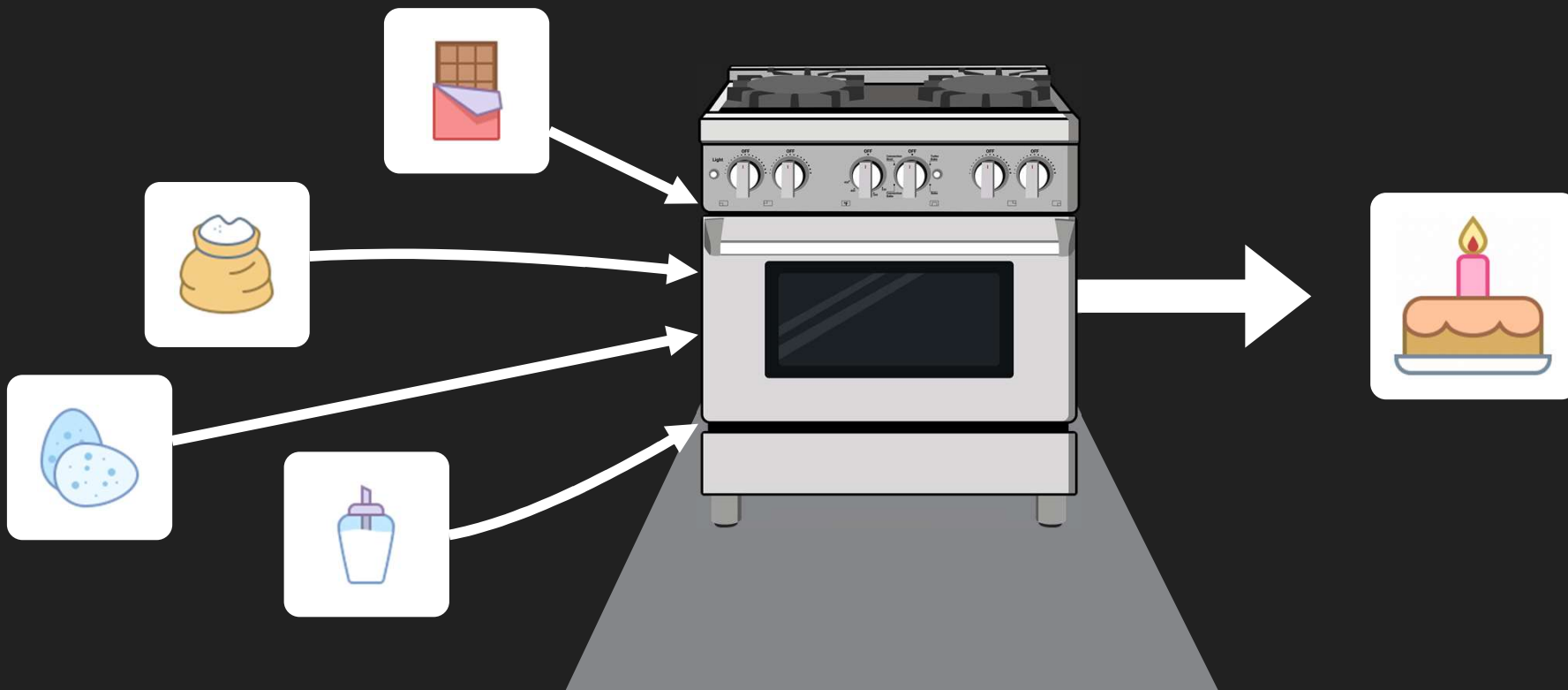
```
let a = 2;  
while(a >= 2) {  
    a = a + 3;  
    /* Aah! 😬 */  
}
```

```
for(let i = 10; i > 0; i++) {  
    /* Aah! 😬 */  
}
```

# Functions

Input:

Output:



# A Very Simple Function

```
function sayHello() {  
    alert("Hello everyone!");  
}
```

# jQuery

jQuery is a JavaScript Library

jQuery greatly simplifies JavaScript programming

jQuery is easy to learn

jQuery is available via:

- CDN (hosted for you):
  - Can guarantee latest version
  - High probability it's already cached on user's web browser
- Locally:
  - Download a specific library version
  - Link as you would any JS file

## Utilising jQuery or Any JS Library

Latest version as of May 2022:

<https://code.jquery.com/jquery-3.6.0.min.js>

Grab the 'minified' version for faster loading:

```
<footer>...</footer>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"
integrity="sha256-/xUj+30JU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4="
crossorigin="anonymous"></script>
<script src="js/script.js"></script>
</body>
</html>
```

# Including jQuery Via CDN



Download the latest version to store in website files:

<https://jquery.com/download>

Place multiple JS files in a 'js' folder — neat and tidy

Grab the 'minified' version for faster loading:

```
<footer>...</footer>
<script src="js/jquery-3.6.0.min.js"></script>
<script src="js/script.js"></script>
</body>
</html>
```

# Including jQuery Via Local

For your **Website Implementation Project**, use local files for any libraries used:

- jQuery can be stored as a JS library for ***Website Implementation*** assessment
- More complex libraries/frameworks/plugins are not encouraged for this course!

## CDN or Local for Me?

# The jQuery Way

```
$(document).ready(function() {  
    /* Something to do when DOM is ready */  
});
```

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

[https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery\\_hide](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hide)



<https://jsfiddle.net/mattluscombe/6fwygsen>

Simpler alternative  
for iterating over  
a set length of values/elements

- Creates variable **for** each iteration
- Separates variable & iterable  
with **of** statement

## For Of/Each Loops



# Events in JavaScript: Click Events

Put `addEventListener()`  <https://jsfiddle.net/mattluscombe/L3tp028b>  
onto a selected DOM/HTML element

- Supports many kinds of events
- Pass along a named/anonymous function `{ }` with code inside to run when event engaged

# Event Listeners



<https://jsfiddle.net/mattluscombe/L3tp028b>

`addEventListener("click")`

is a combination of:

— `addEventListener("mousedown")`

(mouse/trackpad clicked down)

— `addEventListener("mouseup")`

(mouse/trackpad clicked back up)

# Detecting a Click





<https://jsfiddle.net/mattluscombe/L3tp028b>

Named functions are just the custom functions you've made

Anonymous functions negate the need to name the function

# Named vs Anonymous Functions

# Preventing Default Behaviour

```
let readMoreButton = document.querySelector("a#read-more");
readMoreButton.addEventListener("click", function() {

    // Put code here for something to do when this link is clicked

    event.preventDefault();
    // But don't do the normal behaviour.

});
```

# Preventing Default Behaviour

```
let readMoreButton = document.querySelector("a#read-more");
readMoreButton.addEventListener("click", function(event) {

    // Put code here for something to do when this link is clicked

    event.preventDefault();
    // But don't do the normal behaviour.

});
```



Put **event** Argument Here



<https://jsfiddle.net/mattluscombe/L3tp028b>

`this` is a keyword that refers to the currently selected element

# What is `this`?



<https://jsfiddle.net/mattluscombe/L3tp028b>

Use a combination of:

- `addEventListener("mouseenter")`  
(mouse/trackpad hovering on element)
- `addEventListener("mouseleave")`  
(mouse/trackpad has left the element)

# Detecting Mouse Hover



# Events in JavaScript: Keyboard Events



<https://jsfiddle.net/mattluscombe/j8Lmf7cx>

```
addEventListener("keydown")
```

or

```
addEventListener("keyup")
```

# Detecting a Keypress

Use the **Event** object  
to determine the “key”  
from the `keyCode` property



<https://jsfiddle.net/mattluscombe/j8Lmf7cx>

<https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/key>

— This website is a quick way to  
get the codes for certain keys:

<https://keycode.info>

# Determining the Actual Key





# Events in JavaScript: Form Events



<https://jsfiddle.net/mattluscombe/0zkwuyom>

Target the point a form is  
“submitted” with  
`addEventListener(“submit”)`

- More encompassing event than  
just “clicking the submit button”

# Detecting Form Submit

You can also use  
the **Event** object to  
prevent form submissions



<https://jsfiddle.net/mattluscombe/0zkwuyom>

- Useful when you want to stop submission due to badly entered form values

## Preventing Default Behaviour (Again)

Modifying Webpages in JavaScript:

# Adding/Removing Classes



<https://jsfiddle.net/mattluscombe/jL9xcq73>

Changing the `classList` property is more robust than updating “class” attribute

- Adding classes via `classList.add()` will append to pre-existing classes

## Adding Classes



<https://jsfiddle.net/mattluscombe/jL9xcq73>

Removing classes via  
`classList.remove()` will remove the  
specified class, but leave the rest

# Removing Classes



<https://jsfiddle.net/mattluscombe/jL9xcq73>

`classList.toggle()` will check whether the class already exists, then:

- `add()` if it doesn't exist
- `remove()` if it does exist

## Toggling Classes

# A Practical Example

HTML:

```
<a href="popup.html" id="view-popup">View Popup</a>
```



View Popup



Vanilla JS:

```
let viewPopup = document.querySelector("a#view-popup");
viewPopup.addEventListener("click", function(event) {
  document.querySelector("#popup").classList.add("active");
  event.preventDefault();
});
```

jQuery:

```
$("#view-popup").click(function(event) {
  $("#popup").addClass("active");
  event.preventDefault();
});
```



Modifying Webpages in JavaScript:

# Integrating JS Into HTML

# We're Not Using Inline Event Handlers!

```
<a onclick="somethingUseful();" ... </a>
```

```
<form onsubmit="validateTheForm();" ... </form>
```

# Internally

`<script>` Element

`type` Attribute Not Required

```
<footer>...</footer>  
<script>  
  var marksSally = [34, 43, 28, 23];  
</script>  
</body>  
</html>
```

Before `</body>` Is Best

# Externally

`<script>` Element

`type` Attribute Not Required

```
<footer>...</footer>  
<script src="script.js"></script>  
</body>  
</html>
```

Before `</body>` Is Best

`src` Attribute Required

Similar reasons to external CSS:

- Better performance  
(can be cached separately)
- Code consistency across all webpages
- Easier to maintain

# Why External?

# Running JavaScript When the Document's Ready

# Waiting — The Vanilla Way

```
<body onload="initPage();">
```

```
window.onload = function() {  
    /* Something to do when DOM + resources are ready */  
}
```



# A Better Equivalent

```
document.addEventListener("readystatechange", function(event) {  
    if(event.target.readyState == "interactive") {  
        /* Something to do when DOM is ready */  
    }  
});
```



Keep markup, style & behaviour separate:

- HTML in HTML document
- CSS in CSS document
- JS in JS document

Clear separation between the different parts  
of the website's implementation

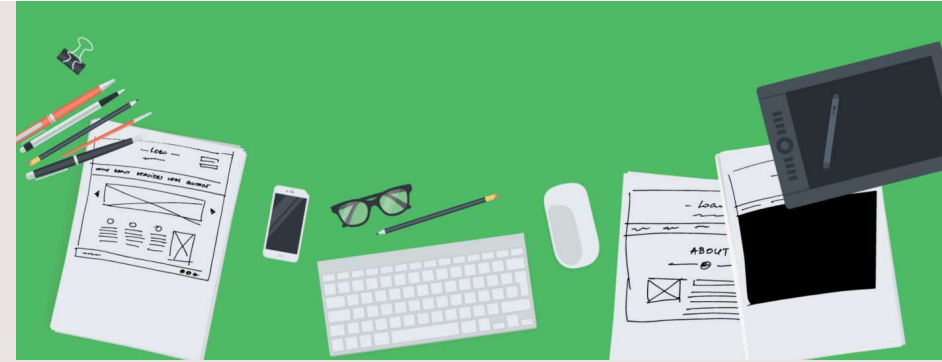
# Good Practice JavaScript

# Human-Centred Design



Principles of Human-Centered Design (Don Norman)  
<https://www.youtube.com/watch?v=rmM0kRf8Dbk>

# Class Activity



Tutorial and Practical activities are available on Blackboard (Week 10) – useful JS/jQuery functions you can use for the implementation assignment

Extra set of self-explanatory slides on Web Accessibility provided by a guest lecturer

*DECO 1400*: Quiz 2 is due on Friday 13<sup>th</sup>

A demo of your website implementation is due next week – attendance is compulsory 😊