DEPARTMENT OF
COMPUTER SCIENCE
計算機科學系

JAVASCRIPT

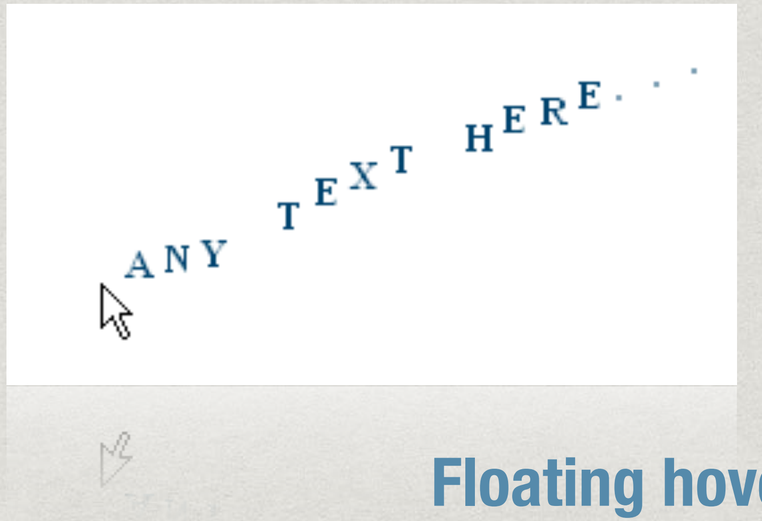
ITEC2016 - DATA-DRIVEN VISUALIZATION FOR THE WEB

CHAPTER 2 - 2018 - HKBU

DR. MARTIN CHOY

JavaScript

- JavaScript was designed to **add interactivity** to HTML pages
- Note that **JavaScript** and **Java** are two completely **different languages** in both concept and design!



Floating hover text

- We use JavaScript to define the **behavior** of a webpage.

What can JavaScript do?

- **React to events**
  - Execute when something happens, like when a page has finished loading or when a user clicks on an HTML element.
  - e.g., **onchange**, **onclick**, etc

```
<script>
    document.getElementById("id").value = "audi"
</script>
```

- **Read and write HTML elements**
  - Read and change the content of an HTML element

Access HTML Element in DOM Tree

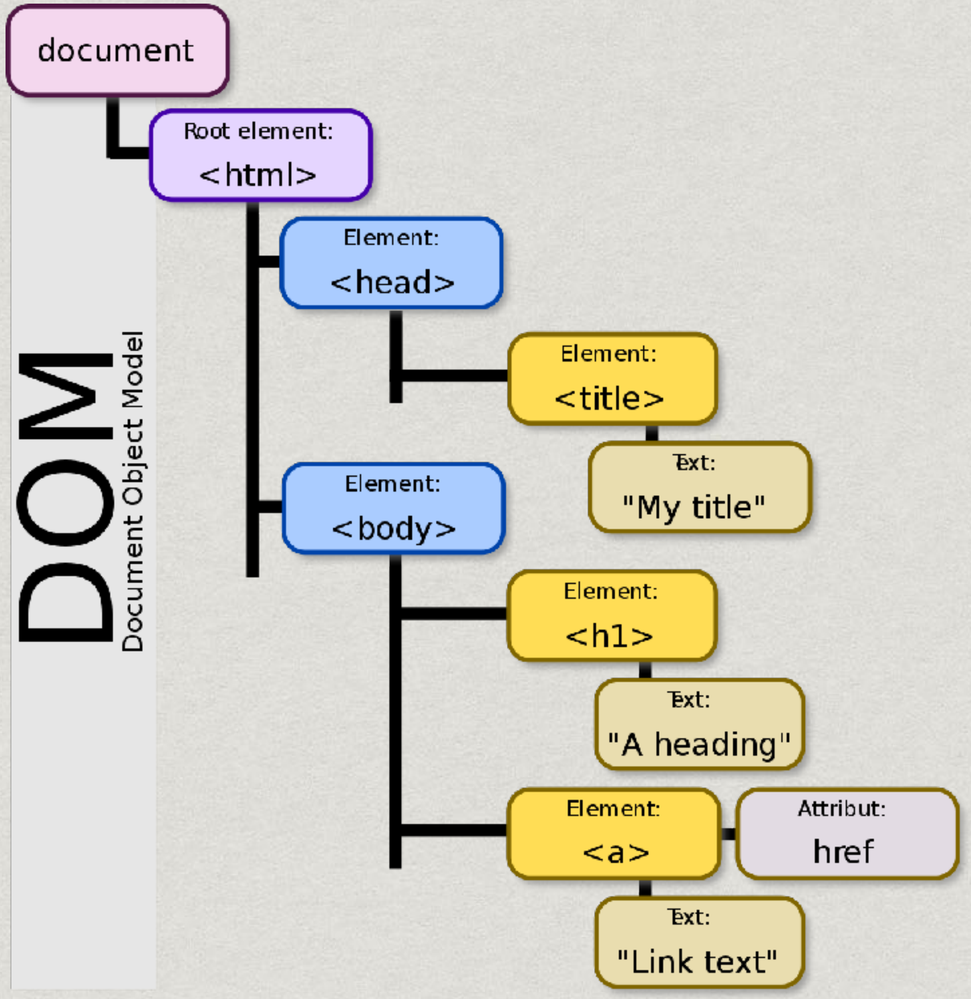
- To get an element **with a specified id**

```
document.getElementById(id)
```

- When a web page is loaded, the browser creates a **Document Object Model (DOM tree)** of the page.

- To access the **parent element** of an arbitrary node (`elem`) in the DOM tree, use:

```
elem.parentNode
```



Read and write HTML elements

- To **set class** to a particular ID element:

```
document.getElementById("myid").className = "active"
```

- To **set an attribute** to the a particular ID element:

```
document.getElementById("myid").setAttribute("style", "display:none");
```

- To **modify the inner HTML**

```
document.getElementById("myid").innerHTML = "Inner content"
```

JavaScript Variables

- Variables are “**containers**” for storing information.
- Loosely typed
- Declare using **var**, e.g., `var x;`
- Naming rules
  - Names are **case sensitive**.
  - Names begin with **a letter** or the **underscore** character.

JAVASCRIPT — LOGIC STRUCTURES

Condition

- In computer, there is only one kind of question you can ask:

  YES / NO **question**

- We call it **boolean test**.
- Examples:
  - 5 equals to 5 → YES
  - 31 is less than 25 → NO
  - 17 is greater than 13 → YES

Syntax of if-statement

- “If” condition can only be evaluated as **either true or false**:
  - when the condition is **true**, the code inside the block(statement) **will execute**;
  - when the condition is false, the code inside the block(statement) **will not execute**.

```
if ( condition ) {
    statement(s)
}
```

Syntax of if-else-statement

- Extend an if block with an **else block**,
  - allowing your program to **choose from two options**.

```
if ( condition ) {
   statement 1
} else {
    statement 2
}
```

- The code inside the else block(**statement 2**) runs when the value of the if block condition is **false**.

Relational Operators

- Now see what is true and false.
- Computer uses **relational operators** to determine conditions

Relational Operators	Meaning
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
!=	not equal
==	equal

Relational Operators

- Here are some examples:

Meaning	Code
variable c is greater than 50	c > 50
variable d is equal to 5	d == 5
variable e is less than or equal to 10	e <= 10
variable f is not equal to 99	f != 99

Relational Operators

- Pay attention to **=** and **==**
- **=** is used to **assign values to variable**
  - eg. “**a=50**” is to give a value(50) to variable(a)
- **==** is used to **test whether left and right sides are equivalent**
  - eg. “**a==50**” is to test whether variable a is equal to 50
- Therefore, **a=50 ≠ a==50**

Looping

Syntax of FOR loop:

```
for (i = 0; i < 10; i = i + 1) {

    document.getElementById("div10").innerHTML += i + " ";

}
```

Output: 0 1 2 3 4 5 6 7 8 9

Looping

Syntax of FOR loop:

```
for ( Initialization ; Condition ; Update ) {

    Statement(s) ;

}
```

Statement:

The code between the braces, { and } , is called a **block**. This is the code that will be repeated on each iteration of the for loop.

Looping

Syntax of FOR loop:

```
for (i = 0; i < 10; i = i + 1) {

    document.getElementById("div06").innerHTML += i + " ";

}
```

Initialization :

- The **variable name i is frequently used**, but there's really nothing special about it.
- We declare the variable i to be 0.

Looping

Syntax of FOR loop:

```
for (i = 0; i < 10; i = i + 1) {

    document.getElementById("div06").innerHTML += i + " ";

}
```

Condition :

Test whether i is **less than 10** or not.

Looping

Syntax of FOR loop:

```
for (i = 0; i < 10; i = i + 1) {

    document.getElementById("div06").innerHTML += i + " ";

}
```

Update :

i will be incremented by 1 each time.

JAVASCRIPT TYPES

JavaScript Types

- 5 language types:
  - Boolean, Number, String, Object and Undefined.
- To check **the type of a variable:** `typeof( variable )`

```
var var1 = true;
var var2 = 1;
var var3 = "One";
```

```
document.getElementById("id1").innerHTML
    = typeof(var1) + "<br><br>" + typeof(var2) + "<br><br>" + typeof(var3);
```

boolean

number

string

Object Type

- An **object** is a special kind of data, with a collection of **properties** and **methods**.
  - Properties are **the values** associated with an object.
  - Methods are **the actions** that can be performed on objects.
- **Car** is the object:

```
var car = {type:"Fiat", model:"500", color: "white"};
```

  - To access the **car's model value**, we can use `car.model`

Array

- An **array** is a special kind of **object**, with a collection of **elements**.
  - An array can **hold many values under a single name**, and you can access the values by **referring to an index number.**
- **vehicles** is the array:

```
var vehicles = ["minibus", "taxi", "motorcycle"];
```

  - To access a particular element of the vehicles, we can use `vehicles[1]`

https://www.w3schools.com/js/js_arrays.asp

The for Statements

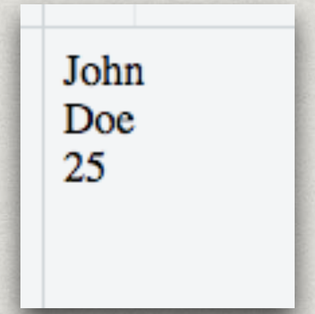
- The **for…in** statement can loop through the **properties of an object**

```
var person = {fname: "John", lname: "Doe", age: 25};

for (var x in person) {

    document.getElementById("id1").innerHTML += person[x] + "<br>";

}
```



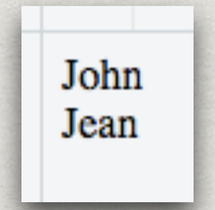
- The **for…of** statement can loop through the **elements of an array**

```
var persons = ["John", "Jean"];

for (var x of persons) {

    document.getElementById("id2").innerHTML += x + "<br>";

}
```



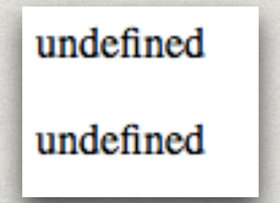
https://jsfiddle.net/gcvv76yj/1/

The Undefined Type

- A variable that **has not been assigned** a value is of type **undefined**.
- A **undeclared variable** is also of type **undefined**.

```
var var1;

document.getElementById("id1").innerHTML
    = typeof(var1) + "<br><br>" + typeof(var2);
```



- Use the following skill to check **whether a variable has been declared,** and avoid **ReferenceError**

```
if (typeof(variable) !== "undefined") console.log("variable exists")
```

Filtering

- Define **a function** to specify what array elements are needed.
  - Pass **this function** as an argument to **the filter() function**.

```
var sukiyakiParticipants = ["Martin", "Kenny", "Jarvis"];

function shorterThan6(value) {
    return value.length < 6;    // counting the number of characters
}

var returnArray = sukiyakiParticipants.filter(shorterThan6);

document.getElementById("div11").innerHTML = returnArray[0];
```

SubString and Split

- We may want to obtain a certain **substring** in a **String**.
- It could be easily done if the substrings are separated by some **delimiters**.
- If so, we can use the **split()** JavaScript function.

```
var str = "How are you doing today?";
var res = str.split(" ");
document.getElementById("demo").innerHTML = res[3];

// doing
```