

JavaScript Flash Cards

Data Types	2
Basic JavaScript Creation	2
Basic JavaScript Terms	2
Basic JavaScript Terms	2
Concatenation	3
Arithmetic Operators	3
Arrays	3
Naming Variable Rules	3
Subject 4	4
If...Else Statements	4
Logical Operators	4
Comparison Operators	4
Subject 4	5
Truthy and Falsy Values	5
Equality & Existence	5
Short Circuit Values	5
Weak or Strong Typed Content	5

Switch Statements	5
Subject 4	6
Functions - declaring a function	6
Immediately Invoked Function	
Expressions (IIFE) p 97	6
Loop Types	6
Loops	6
Subject 4	7
Subject 1	7
Event Flow (chapter 6 Events	
continued)	7
Events (chapter 6)	7

Basic JavaScript Terms

Object

Property

Name: Value (Value Pairs)

Method()

Event

JavaScript scripts reside alongside html document, recommended in a separate folder for organization.

Basic JavaScript Terms

JavaScript appears in html as
<script src="dir/filename.js"></script>

Basic JavaScript Creation

```
var name = variable value;  
code (parameters) {code  
} {code  
} {code  
}
```

`Document.methodname(parameter, parameter)`

Data Types

Numeric - numbers, including floating values.
String - 'letters and other characters.'
Boolean - will return **true** or **false**.

Naming Variable Rules

1. Name must begin with letter, \$, or _.
2. Name cannot use a - or .
3. Cannot be a keyword or reserved word.
4. All variables are case sensitive.
5. Be correctly descriptive.
6. Use camelCase for new words, and - (dashes) are also acceptable.

Arithmetic Operators

+ Addition	- Subtraction
/ Divison	* Multiplication
++ increment	- +1 from current number
-- Decrement	- -1 from current number
% Modulus	- divides by value and returns the remainder.

Arrays

Create by - (below is an array literal)
var colors;

colors = ['white', 'black', 'custom'];

Call by -

colors.item(0) (returns 'white')

OR

colors[2] (returns 'custom')

Arrays began at 0.

Concatenation

+ combines values to a string if a string value is present in the concatenation.

Comparison Operators

`==` equal to
`!=` is not equal to
`===` strict equal to
`!==` strict not equal to
`>` greater than
`<` less than
`>=` greater than or equal to
`<=` less than or equal to

If...Else Statements

```
keyword (condition) {  
    code to execute if true();  
}  
else {  
    code to execute if false();  
}
```

Logical Operators

`&&` logical and (tests more than one condition)
`||` logical or (tests at least one condition)
`!` Logical not (takes a single Bool and inverts it)

Subject 4

Content

Switch Statements

```
switch (var being evaluated) {  
  case (var result):  
    var if true;  
    break;  
  case (next until choose to stop):  
    var if true;  
    break  
  default:  
    var if none resolve true;  
    Break;  
}  
  
element.element = var resolved as true
```

Truthy and Falsy Values

See p 167

Equality & Existence

See p 168

Short Circuit Values

See p 169

Weak or Strong Typed Content

Weak Typing = data type can change.
Strong Typing = data type for a variable
is specified prior.

Subject 4

Content

Loops

```
For (var i = value; I < 10; i++) {  
  |      /      /      \  
keyword  initialization  condition  update  
  document.write(i);  
    \  
    code to execute during loop
```

Loop Types

```
for  
while  
do
```

Functions - declaring a function

```
Function function name(argument1, argument2) {  
  code;  
  return var;  
}
```

Immediately Invoked Function Expressions (IIFE) p 97

A way to "brute force" a piece of code.
"Wrappers" to make things happen.

Subject 4

Content

Events (chapter 6)

Events are raised or fired (likely by a user), that then triggers a function or script.

Pp 246-247 outline many of the events.

How you script an event uses the DOM and is like other Javascript DOM manipulation

```
Element.addEventListener - functionName;
```

```
^^Event handler format;
```

```
element.addEventListener('event',functionName [,Boolean]);
```

Subject 1

Content

Event Flow (chapter 6 Events continued)

event bubbling: starting at most specific to least.

Event capturing: event starts at least specific and flows inward.

User Interface Events (pp 272-282)

These pages address the details of having JavaScript interact with what is going onscreen, what the user is doing. Events like focus and blur, mouse clicks, keyboard, events... all controllable by events and event listeners via DOM manipulation.

Subject 4

Content