Web Development with HTML, CSS, and Javascript

Set up
Install Node.js http://nodejs.org/   Test it with: console.log("Hello, World!");
Install Editor: https://atom.io/
1.   Click on **Packages** --> **Commmand Palette** --> Select **Toggle**.
2.   Type **Install Packages and Themes**.
3.   Search for **Color picker** and then install it
gitbub repo: https://github.com/bhou14/webdev

1.   Introduction

First Page:
```
<html>
<head><title>My First Page</title></head>
<body>
<p>Helle, there</p>
</body>
</html>
```
Add Style:
```
<style>
p {color:blue; font-family: cursive; font-size: 3.0em;}
</style>
```


Add Script:
```
<html>
<head><title>My First Page</title>
<style>
p {color:blue; font-family: cursive; font-size: 3.0em;}
</style>
<script type="text/javascript">
    var myObj = null;
    function init(){
      myObj = document.getElementById('myobj');
      myObj.style.position= 'relative';
      myObj.style.left = '0px';
    }
    function moveRight(){
      myObj.style.left = parseInt(myObj.style.left) + 100 + 'px';
      setTimeout(moveRight, 3000);
    }
    window.onload =init;
</script>
</head>

<body>
```

```html
<p id="myobj">Hello, there!</p>
<input type="button" value="Click Me" onclick="moveRight();" />
</body>
</html>
```

Putting it together:
```html
<html>
<head><title>My First Page</title>
<link rel="stylesheet" href="style.css">
<script src="app.js"></script>
</head>

<body onload="init();">
<p id="myobj">Hello, there!</p>
<input type="button" value="Click Me" onclick="moveRight();" />
</body>
</html>
```

Common HTML Tags:

\<h1\> … \<h6\>	— header text
\<p\>	— Paragraph
\<div\>	— division, a block of content
\<span\>	— similar to div, but doesn't force a line-feed
\<img\>	— image
\<a\>	— link
\<table\>\<tr\>\<td\>	— table, row, and column
\<ul\>\<li\>	— list

Common CSS Selectors:

Selector	Example	Description
.class	.intro	Selects all elements with class="intro"
#id	#firstname	Selects the element with id="firstname"
element	p	Selects all \<p\> elements
element,element	div, p	Selects all \<div\> elements and all \<p\> elements
element element	div p	Selects all \<p\> elements inside \<div\> elements
element>element element	div > p	Selects all \<p\> elements where the parent is a \<div\>
element+element after \<div\> elements	div + p	Selects all \<p\> elements that are placed immediately

2.   Javascript Basics

Define variables:
```javascript
var myString = "some text";        // a string
```

```javascript
var myInt = 12;                    // a integer
var isOk = true;                   // boolean
```

JavaScript Array & Object
```javascript
var arr = ["one", "two", "three"];
var first = arr[0];
var arr2 = new Array();
arr2[0] = "one";
arr2[1] = "two";
arr3[2] = "three";
var arr3 = new Array();
arr3.push("one");
arr3.push("two");
arr3.push("three");

var obj = {
   name: "My Object",
   value: 7,
   type: "Object Type",
   getType: function() { return this.type; }
};
console.log(obj.name);
console.log(obj["value"]);
console.log(obj.getType());
```

Operators:
```
  arithmetic: +  -  ++  --  *  /  %
  assignment: =  +=  -=  *=  /=  %=
  comparison: ==  ===  !=  !==  >  >=  <  <=
            &&  ||  !
```
Conditions:
```
  if(...) { ... }
  else if(...) { ... }
  else { ... }
```

3.   Function

Declare a function:
```javascript
function myFunc(param1, param2) {
...
}
var myFunc = function(param1, param2) {
  var x = param1 + param2;
  return x;
}
```

Calling a function:
```
myFunc('this', 'that');
var ret = myFunc('second', 'third');
```

Passing function as a parameter:
```
AnotherFunc(p1, p2, myFunc) {
  return myFunc(p1, p2);
}
```

4.   Loops

```
var i = 1;
while (i<5){
  console.log("Iteration " + i);
  i++;
}
```

```
for(var i = 0; i<10; i++)
{
   console.log("iteration " + i);
}
```

```
var days = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday"];
for (var idx in days){
  console.log("It's " + days[idx]);
}
```

```
var days = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday"];
for (var idx in days){
        console.log("It's " + days[idx] );
}
```

6.   Node.js Server

```
var express = require('express');
var app = express();

app.use('/', express.static('./static'));
//      use('/images', express.static('../images')).
app.listen(8088);
```