



introduction to media computing week 11

Today's topics (week 11)



- Review of function Part 1
- Function Part 2

Review: Function Part 1

js

We have been using various p5.js built-in 'functions' in our sketches, they include:

```
noFill();  
noStroke();  
...
```

1. Simple functions

```
point(10,10);  
stroke(255);  
...
```

2. Functions which take parameters

```
let y = floor(x);  
let p = lerp(a,b,0.1);  
...
```

3. Functions which take parameters, and return values

Reveiw: Function Part 1

js

1

drawing instructions
that we use repeatedly.

```
stroke(0);  
strokeWeight(5);  
noFill();  
rect(10,10,100,100);
```

2

Make that block as
a simple function
named `myStyle()`.

```
function myStyle() {  
  stroke(0);  
  strokeWeight(5);  
  noFill();  
}
```

3

Substitute the block
by our new
`myStyle()`.

```
myStyle();  
rect(10,10,100,100);
```

Function: Write our own simple function

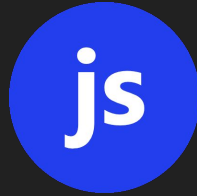
js

```
function <name>() {  
    // code here  
}
```

Function: function with parameters

js

```
function <name>(param1,param2,...) {  
    // code here  
}
```



Function Part 2

Function: function which returns result

js

```
function <name>(param1,param2,...) {  
    let result;  
    // code to compute result  
    return result;  
}
```


Function: function which returns result

js

```
function <name>(param1,param2,...) {  
    let result;  
    // code to compute result  
    return result;  
}
```

Function: function which returns result

js

Example 1

```
function smallerOne(p1,p2) {  
    let result;  
    if (p1 <= p2) {  
        result = p1;  
    }  
    else {  
        result = p2;  
    }  
    return result;  
}
```

Function: function which returns result

js

Example 2

```
function maxMember(array) {  
    let result = array[0];  
    for (i = 1; i < array.length; i++) {  
        if (array[i] > result) {  
            result = array[i];  
        }  
    }  
    return result;  
}
```

Function: Example 2

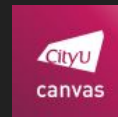
js

```
1 function maxMember(array) {  
2   let result = array[0];  
3   for (i = 1; i < array.length; i++) {  
4     if (array[i] > result) {  
5       result = array[i];  
6     }  
7   }  
8   return result;  
9 }  
10  
11 function setup() {  
12   createCanvas(400, 400);  
13   let myArray = [10, 9, 7, 5, 13, 4, 2];  
14   console.log(maxMember(myArray));  
15 }
```

Console

Clear ▾

13



```
function maxMemberIndex(array) {  
  
}  
  
function setup() {  
  let a = [12,3,40,22,0,1,7];  
  console.log(maxMemberIndex(a));  
}
```

Complete the function

`maxMemberIndex()`

such that it returns the
'index' of the largest
number of a given array.

Expected output:
Your code should return
the index no. 2.



```
function oddIndexMembers(array) {  
  
}  
  
function setup() {  
  let a = [12,3,40,22,0,1,7];  
  console.log(oddIndexMembers(a));  
}
```

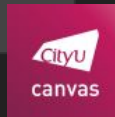
Complete the function

`oddIndexMembers()`

such that it extracts and returns the members with an 'odd' index number of the input array.

Hint: You will need to use the `%` operator and `.push()` function for array.

Expected output:
Your code should return this array: `[3,22,1]`



```
function evenNumbers (array) {  
    
}  
  
function setup() {  
  let a = [12,3,40,22,0,1,7];  
  console.log(evenNumbers(a));  
}
```

Complete the function

`evenNumbers()`

such that it extracts and returns the members which are even numbers of the input array.

Hint: You will need to use the `%` operator and `.push()` function for array.

Expected output:
Your code should return this array: `[12, 40, 22, 10]`

