



FEWD - Functions

Thomas Broomfield

Technical Lead, PwC Ventures.

Agenda

- Food
- Review
- Anonymous Functions
- Practice!

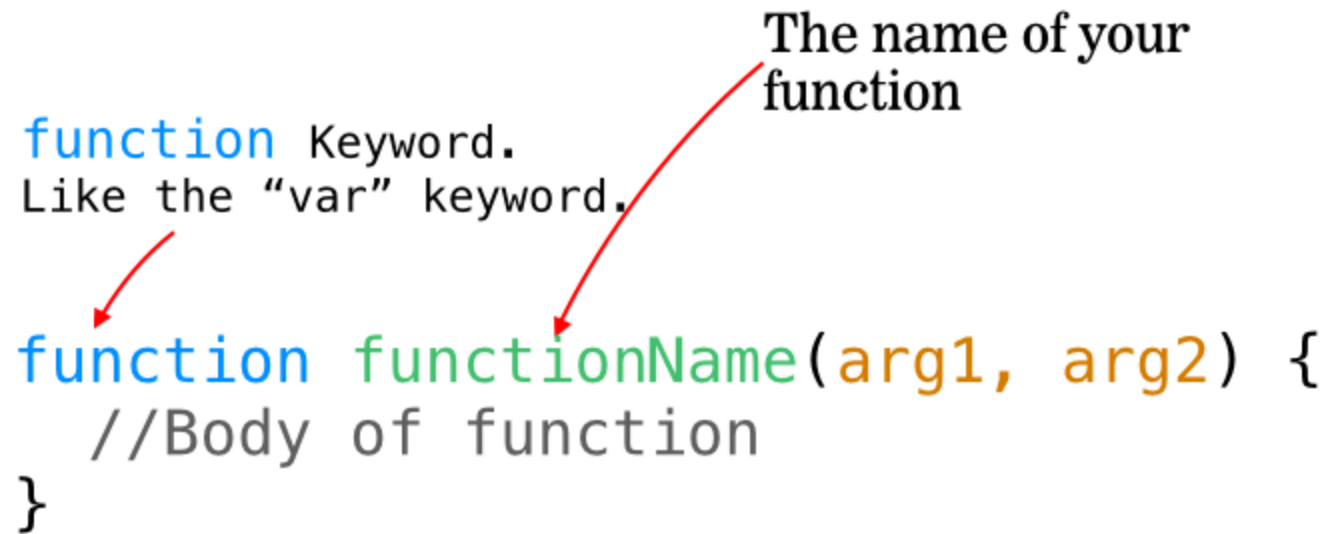
Review

Functions - Review

##Functions Syntax

`function` Keyword.
Like the “var” keyword.

The name of your
function




```
function functionName(arg1, arg2) {  
    //Body of function  
}
```

The diagram illustrates the syntax of a function declaration. A red arrow points from the text 'function Keyword. Like the “var” keyword.' to the word 'function' in the code. Another red arrow points from the text 'The name of your function' to the word 'functionName' in the code. The code itself is color-coded: 'function' is blue, 'functionName' is green, and 'arg1' and 'arg2' are orange.

Function Calls

```
function helloWorld() {  
  console.log("Hello Functions");  
}
```

```
helloWorld(); //Prints "Hello Functions to the  
console.
```



The brackets execute the function.
Try calling the function without
them to see what happens.

Function Arguments

Arguments let you pass data into the function

```
function functionName(arg1, arg2) {  
    //Body of function  
}
```

A diagram with two red arrows. One arrow starts from the text 'Arguments let you pass data into the function' and points to the arguments 'arg1, arg2' in the function signature. The second arrow starts from the text 'The functions executed code goes between the { } brackets. Much like an “if” statement.' and points to the body of the function, specifically the line '//Body of function'.

The functions executed code goes between the { } brackets. Much like an “if” statement.

Function Arguments

```
function addAndPrint(num1, num2) {  
    var sum = num1 + num2;  
    console.log(sum);  
}
```

```
addAndPrint(1, 2); // Result is 3
```

```
addAndPrint(8, 2); // Result is 10
```


Return Functions

When you use the `return` keyword in a function. It sets an explicit output.

For instance:

```
function square(x) {  
  return x * x  
}  
  
var foo = square(3);
```

What will foo equal?



Cash Register

Anonymous Functions

Using our example before:

```
x = 3  
square(x) // 9  
square(3) // 9
```

It doesn't matter if we pass in a variable which is equal to something, or that thing itself, it's the same result.

We could say that the `3` on the last line is anonymous - it doesn't have a name.

Anonymous Functions

We can do the same with functions:

```
$(document).ready(start);  
  
function start() {  
    // Do stuff  
}
```

Is exactly the same as:

```
$(document).ready(function() {  
    // Do Stuff.  
});
```

Anonymous Functions

Or, with a jQuery click:

```
$( '.button' ).ready(doThing);  
  
function doThing() {  
    // Do stuff when button is clicked.  
}
```

Is the same as:

```
$( '.button' ).ready(function() {  
    // Do stuff when button is clicked.  
});
```

Blackout!

- Write a program that has a single button.
- When you click that button once, the entire page should turn black.
- When you click it again, the page should return to white.
- For extra points:
 - Make it so the background fades instead of going immediately.
 - Make it so it cycles through a set of colours, instead of just black and white.

