Exercise: Functions

Answer:

```javascript
function showTip(amount, tipRate) {
    var tip;
    tip = amount * tipRate;
    document.write("the tip is: "+ tip);
}
```

Exercise: Functions

Modify this function so that it returns the tip rather than outputting it into an HTML document (as done in the function below):

```javascript
function showTip(amount, tipRate) {
    var tip;
    tip = amount * tipRate;
    document.write("the tip is: "+ tip);
}
```

Exercise: Functions

Solution:

```javascript
function calculateTip(amount, tipRate) {
    var tip;
    tip = amount * tipRate;
    return(tip);
}
```

(We’ve renamed the function to reflect better what it is doing.)

Selected RQs

- Does the word "return" mean equal in the function when we design a JavaScript?

```javascript
function gimmeFive() {
    return 5;
}
```

(submitted by Xin – 2011W1 Student)
Selected RQs

- Does the word "return" mean equal in the function when we design a JavaScript?

```javascript
function gimmeFive() {
    return 5;
}
```

(Submitted by Xin – 2011W1 Student)

Anatomy of a Javascript Function

**Watch for syntax:**
- the parentheses in the declaration (must be present even if the function has no parameters!)
- the `{` and `}` around the body
- the semicolons (at the end of each statement, optional after the return statement)

```javascript
function calculateTip(amount) {
    var tip = amount * 0.12;
    return tip;
}
```

Selected RQs

Why are variables categorized by local or global? Does it matter if the variable could automatically apply to all of the sprites, since we still have the option of not actually applying it?

(Submitted by Lisa)

In the Scratch tutorial, it mentioned that a variable can be local or global. I understand how that works for Scratch, but how will that appear on a regular programming language like JavaScript?

(Submitted by Norma)
Local vs Global Variables

```javascript
function calculateTip(amount, tipRate) {
    var x;
    x = amount * tipRate;
    return(x);
}
// variable x is local to func. calculateTip
```

```javascript
var x;
function calculateTip(amount, tipRate) {
    x = amount * tipRate;
    return(x);
}
// variable x is global
```

Clicker question: Local vs Global

Which is better?

(A) var x;

```javascript
function calculateTip(amount, tipRate) {
    x = amount * tipRate;
    return(x);
}
```

(B) function calculateTip(amount, tipRate) {
```
    var x;
    x = amount * tipRate;
    return(x);
```

(C) neither – they are equally good

Functions: Things to remember

- Using a function requires two parts: **declaration** (where we define the function) and **call** (where we execute it).

- Parameters can be seen as special variables that are set to certain values when the function is called and before the body is executed. They are used to pass data into a function.

- Variables declared inside a function are private to that function and cannot be accessed outside of it.
Functions: Things to remember

- A return statement is used to pass data (a result) back from the function to the place where it was called.

- After a return statement is executed, execution of the function stops immediately (even if further statements follow in the function declaration)!

- Functions can call other functions (or even themselves – but be very careful with this, since it can lead to infinite regress)

Functions: Things to remember

- Functions may not have parameters (if they do not require input data).

- Functions may not declare variables.

- Functions may not have a return statement (if they do not return data back to the calling process).

- Functions may have more than one return statement (but only one of them gets ever executed in any given call).

Selected RQs

In the “JavaScript Primer – Functions” states that a function can only return one value at a time. Why is this? I tried it out, and it returns only the first value that was set out: why the first rather than the second? Why not just return an error message? If you wanted to perform both those tasks, would you just create a second function? That same section compares this illegal action to the fact that a variable can only contain one value at a time. If you defined a variable as two things, (e.g. Line 1: var age=5; Line 2: var age=9) it would just replace the original value/redefine the variable “age” as the second value, right?

(submitted by Emily)

Clicker question: Variables

What is the value of age after executing

```javascript
var age;
age = 5;
age = 9;
```

A. 5
B. 9
C. undefined/error
Clicker question: Variables, Functions

• What is the value of \( y \) at the end of \( \text{confuse}(10) \)?

A. 10  
B. 20  
C. 30  
D. 40

```javascript
function confuse(y) {
    var x;
    x = 2 * y;
    y = y + x;
    return (x);
}
```

• What value is returned by \( \text{doubleConfuse}(3,10) \)?

A. 3  
B. 13  
C. 23  
D. 33

```javascript
function confuse(y) {
    var x = 2 * y;
    y = y + x;
    return x;
}

function doubleConfuse(a,b) {
    var x;
    x = a + confuse(b);
    return x;
}
```

Selected RQs

For the upcoming lab, I noticed that the javascript code is included in the head of the HTML page whereas the HTML code, including the form, is included in the body of the HTML page. I don't understand how the javascript in the head section can refer to variables defined in the body of the page, which is coded further down. I thought that the computer would read the code from top to bottom. In this case, how can it process information inputted in the form when it is referred to later in the code? (For example, in the Battleship game, a function defines a variable based on the user input which is garnered further down in the code).

(submitted by Kelsie)

Clicker question: Conditionals

• What value of \( x \) after executing the following?

\[
x = -10;
\]
\[
\text{if} \ (x < 0) \ \{
    x = x * 1.195;
\} \ \text{else} \ \{
    x = x * 1.015;
\}
\]

A. -11.95  
B. -10  
C. -10.15  
D. 0
Conditional Statements

Let's revisit our CalculateTip function:

```javascript
function CalculateTip(amount, tipRate) {
    tip = amount * tipRate;
    return (tip);
}
```

What if we want to determine the tip amount based on service?
If service is "good" then give a good tip...

Determine the tip amount:

```javascript
if (serviceLevel > 7)  {
    // scale of 1..10
    adjustedRate = tipRate + 0.03;   // +3%
} else {
    adjustedRate = tipRate;  // no extra tip
}
```

Exercise: Functions, Conditionals

- What if we want to have 3 different levels of tip? Good / Normal / Terrible ?

- Write out the new function in full (parameters, return value, etc.)

- Write down a first solution individually, then compare your solution with a neighbour! Those with laptops, try it out! (Google: tryit javascript)

One solution:

```javascript
function calcFlexTip(amount, baseRate, serviceLevel) {
    var adjustedRate;
    if (serviceLevel > 7) {
        adjustedRate = baseRate + 0.03;   // +3%
    } else if (serviceLevel < 3) {
        adjustedRate = baseRate * 0.5;  // half rate
    } else {
        adjustedRate = baseRate;  // no extra tip
    }
    tip = amount * adjustedRate ;
    return (tip);
}
```
Anatomy of a Conditional Statement

Logical condition (= something that must be true or false)

```java
if (serviceLevel == 10) {
    adjustedRate = baseRate * 2;
} else if (serviceLevel < 3) {
    ...
} else {
    ...
}
```

Conditional statements can be nested (one inside another)

Clicker question: Conditionals

- What value of \( x \) after executing the following?
  
  ```java
  x = -10;
  if (x = 0) {
      x = x * 1.195;
  } else {
      x = x * 1.015;
  }
  ```

  A. -11.95  
  B. -10.15  
  C. -10   
  D. 0