Functions and more

Finishing Week 7
Class Resources

- Fz
- Notepad++
- Visual Studio Code
- Atom
- Bootstrap
- W3schools
- Codecademy
- Visualize Python Tutor
- Paletton
- Adobe Color
VISUALIZE CODE AND GET LIVE HELP

Learn Python, Java, C, C++, JavaScript, and Ruby

Python Tutor, created by Philip Guo (@pgbovine), helps people overcome a fundamental barrier to learning programming: understanding what happens as the computer runs each line of code.

Write code in your web browser, see it visualized step by step, and get live help from volunteers.

Related services: Java Tutor, C Tutor, C++ Tutor, JavaScript Tutor, Ruby Tutor

So far, over five million people in over 180 countries have used Python Tutor to visualize over 75 million lines of code, often as a supplement to textbook learning and online tutorials.
Write code in JavaScript ES6 ▼

1  var x = 10;
2  var y = 20;
3  var z = x + y;
<table>
<thead>
<tr>
<th>Frames</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global frame</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>undefined</td>
</tr>
<tr>
<td>y</td>
<td>undefined</td>
</tr>
<tr>
<td>z</td>
<td>undefined</td>
</tr>
</tbody>
</table>
1 var x = 10;
2 var y = 20;
3 var z = x + y;

Edit this code
```javascript
1 var g = 210;
2 var t = 20;
3 var z = g/t;
```

**Frames**

<table>
<thead>
<tr>
<th>Global frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>g</td>
</tr>
<tr>
<td>t</td>
</tr>
<tr>
<td>z</td>
</tr>
</tbody>
</table>

**Objects**

Edit this code
```javascript
1 var g = 210;
2 var t = 20;
3 var z = Math.floor(g/t);
```

This code has just executed.

The following objects are defined:

<table>
<thead>
<tr>
<th>Global frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>g</td>
</tr>
<tr>
<td>t</td>
</tr>
<tr>
<td>z</td>
</tr>
</tbody>
</table>

Edit this code to set a breakpoint; use the Back and Forward buttons to jump there.
```javascript
1 var g = 210;
2 var t = 20;
3 var z = Math.floor(g/t);
4 var x = g*t;
```

**Frames**
- Global frame
  - g: 210
  - t: 20
  - z: 10
  - x: 10

*Edit this code*

Set a breakpoint; use the Back and Forward buttons to jump there.
```javascript
1  var g = 7;
2  var t = 2;
3  var z = Math.floor(g/t);
4  var x = g%t;
```

**Edit this code**

**Frames**

```
Global frame

g  7
t  2
z  3
x  1
```

**Objects**

Executed code:

- Place a breakpoint; use the Back and Forward buttons to jump there.
\[
f(x) = \frac{1}{2}x + 2;
\]
\[
x = 1 \Rightarrow f(x) = \frac{3}{2}
\]

Even if your function has no parameters, you still need an opening and closing set of parens, like \((\).\]

Always start with the keyword “function”.

Follow the function keyword with the name of your function.

And then zero or more comma separated parameters between parentheses.

The body sits between two curly braces and contains a set of statements (just like the statements you’re used to).

The return statement includes an expression, which is returned as a result of calling the function.

A function can include a statement with the return keyword, but it doesn’t have to.
\[ y = f(x) = x + 2 \]

Write code in JavaScript ES6 ▼

```javascript
var myLevel = 7;
var myScore = 2;

function addScore (level, score) {
    var bonus = level * score * .1;
    return score + bonus;
}

var myResult = addScore(myLevel, myScore);
```
1 var myLevel = 7;
2 var myScore = 2;
3
4 function addScore (level, score) {
5    var bonus = level * score * .1;
6    return score + bonus;
7 }
8
9 var myResult = addScore(myLevel, myScore);
```javascript
var myLevel = 7;
var myScore = 2;

function addScore (level, score) {
    var bonus = level * score * .1;
    return score + bonus;
}

var myResult = addScore(myLevel, myScore);
```

Global frame:
- `myLevel`: 7
- `myScore`: 2
- `addScore`: undefined
- `myResult`: 2.2

---

Function `addScore`:
- `level`: 7
- `score`: 2
- `bonus`: 1.4
- `score + bonus`: 2.6

---

Edit this code

- The line that has just executed:
- The line to execute next:
- Set a line of code to set a breakpoint; use the Back and Forward buttons to jump there.
var myLevel = 7;
var myScore = 2;

function addScore (level, score) {
  var bonus = level * score * .1;
  return score + bonus;
}

var myResult = addScore(myLevel, myScore);
```javascript
var myLevel = 7;
var myScore = 2;

function addScore (level, score) {
    var bonus = level * score * .1;
    return score + bonus;
}

var myResult = addScore(myLevel, myScore);
```

### Frames
```
<table>
<thead>
<tr>
<th>Global frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>myLevel</td>
</tr>
<tr>
<td>myScore</td>
</tr>
<tr>
<td>addScore</td>
</tr>
<tr>
<td>myResult</td>
</tr>
</tbody>
</table>
```

### Objects
```
function addScore(level, score) {
    var bonus = level * score * .1;
    return score + bonus;
}
```
onclick = "somefunc()";
```javascript
var useThisOperator = "";
function setOperator(selectedOperator){
    useThisOperator = selectedOperator;
}
```
var correctAnswer = eval(operand1 + useThisOperator + operand2);
function bark(dogName, dogWeight) {
  if (dogWeight <= 10) {
    return dogName + " says Yip";
  } else {
    return dogName + " says Woof";
  }
}

var sound = bark("Fido", 50);
alert(sound);
Photoshop Assignment

COGS3 – Introduction to Computing
How to Create a Glazed Cookie Text Effect in Adobe Photoshop

Use Photoshop's brush settings, layer styles, and adjustment layers to create a fun, bright glazed cookies text effect.
Jenny Le

I'm a self-taught artist from Vietnam who loves all kinds of photo manipulation. If you want to see my other works and tutorials, you can follow me on Facebook, Behance, Instagram or watch me on DeviantArt.

jennyle88.deviantart.com/
40 Examples of stunning Cinemagraphs

by Dustin Betonio | Last updated Jan 21, 2017 | Photography | 44 comments
Bootstrap Carousel Plugin

The Carousel Plugin

The Carousel plugin is a component for cycling through elements, like a carousel (slideshow).

Tip: Plugins can be included individually (using Bootstrap's individual "carousel.js" file), or all at once (using "bootstrap.js" or "bootstrap.min.js").

Carousel Example
Add `<div class="carousel-caption">` within each `<div class="item">` to create a caption for each slide:

```html
<div id="myCarousel" class="carousel slide" data-ride="carousel">
   <!-- Indicators -->
   <ol class="carousel-indicators">
      <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
      <li data-target="#myCarousel" data-slide-to="1"></li>
      <li data-target="#myCarousel" data-slide-to="2"></li>
   </ol>

   <!-- Wrapper for slides -->
   <div class="carousel-inner">
      <div class="item active">
         <img src="la.jpg" alt="Chania">
         <div class="carousel-caption">
            <h3>Los Angeles</h3>
            <p>LA is always so much fun!</p>
         </div>
      </div>
   </div>
</div>
```

*you need to have captions*