Why do app developers have such high insurance rates?

Because they are always crashing.


Thank you, Kai
Last Lecture

- Investigated different kinds of functions

- Looked at the execution flow when Python executes our functions using the **Python Code Visualizer**
**Review: Different kinds of functions**

**Activity:** Fill this table using the four functions we created last lecture.

<table>
<thead>
<tr>
<th>Function that takes argument(s)</th>
<th>Function that does not take argument(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function that returns a result (i.e., a returned value)</td>
<td></td>
</tr>
<tr>
<td>Function that does not return a result (i.e., a returned value)</td>
<td></td>
</tr>
</tbody>
</table>
Review - Your turn!

- **Problem Statement:**
  - Create a function that prints a character x number of times in a row

- **Requirement:**
  - Let’s not forget about the **Generalization Principle**
Review - Function

• Why create functions?
  • Functions make our program easier …
    • To read
    • To implement and modify
    • To test and debug
    • To re-used in other programs
  • Functions make our program shorter …
    • Eliminate repeated code fragments

• Three guidelines to follow when creating functions
  • Descriptive name of function
  • Parameter(s) or not
  • Returns a value or not
Today’s Menu

• We shall introduce another field of Computing Science:
  • Graphics and Animation
• We shall introduce a fun Python module:
  • Turtle
Graphics and Animation

• Pixar movies and your favourite 3D animated films these days are built with code

• Indeed, **animated movies** and **video games** today rely heavily on 2D and 3D graphics and animation technology

• In this unit, we’ll make basic animations using code, and learn how to build complex and beautiful graphics
1. **Computing Science**
   - In this unit, we’ll learn about the computing science field of **graphics and animation**

2. **Algorithm**
   - One of the algorithms we shall learn is called **recursion**

3. **Python**
   - We’ll learn about **Turtle** graphics, and continue learning about **functions and loops**
Let’s try it!

- **Problem Statement:**
  - Write a program that draws a blue square with a **Turtle**
  - First, we need to know the **Turtle** coordinate system:
How about this!

- **Step 1 - Problem statement:**
  - Write a program that draws a chocolate chip cookie with our Turtle

- **Step 2 - Design**
  - As a software developer, I do not need to be told to create a function 😊
  - I know that the functionality of **drawing a cookie** needs to be implemented as a function
But what if I want many cookies?

- Solution:
  - Could I copy and modify the code many times, each instance of the copied code drawing one cookie?

- Hum… We now know that this solution would lead to a lot of repeating code, which is not a good idea!

Why?
But what if I want many cookies?

- **Step 2 - Design**
  - Should I use a `for` loop?
Review - For next lecture

1. How do we create a Turtle object, given the turtle module?

2. What is the keyword necessary to make a while loop?

3. What is the keyword necessary to create a function?

4. What is the difference between function parameters and function arguments?

5. Why do we place our functions at the top of our program?
Next Lecture

• More turtle and functions