


The Beauty and Joy of Computing
Lecture #4 : Functions

UC Berkeley EECS Sr Lecturer SOE Dan Garcia

Quest (first exam) in 16 days!!

THE FUTURE OF VIDEO GAMES?

Valve (video game makers of Half-Life) believes the future of video games may not be in the *input* device (ala the Wii remotes or your body ala Kinect), but the *output* device! What is shown on the right is an *augmented reality* device, layering 3D content onto the real world.




<http://www.nytimes.com/2012/09/09/technology/valve-a-video-game-maker-with-few-rules.html>

Generalization (in CS10) REVIEW

- You are going to learn to write functions, like in math class:

$$y = \sin(x)$$

- sin is the function
- x is the input
- It returns a single value, a number

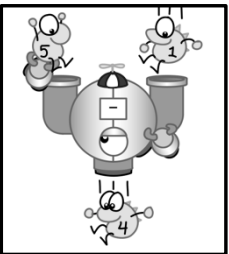


"Function machine" from *Simply Scheme* (Harvey)

UC Berkeley "The Beauty and Joy of Computing" : Functions (3)

Function basics

- Functions take in 0 or more inputs and return exactly 1 output
- The same inputs **MUST** yield same outputs.
 - Output function of input only
- Other rules of functions
 - No state (prior history)
 - No mutation (no variables get modified)
 - No side effects (nothing else happens)



CS Illustrated function metaphor

UC Berkeley "The Beauty and Joy of Computing" : Functions (5)

More Terminology (from Math)

- Domain**
 - The "class" of input a function accepts
- Range**
 - All the possible return values of a function
- Examples**
 - Sqrt of
 - Positive numbers
 - Length of
 - Sentence, word, number
 - `_ < _`
 - Both: Sentence, word, number
 - `_ and _`
 - Booleans
 - Letter `_` of `_`
 - Number from 1 to input length
 - Sentence, word, number

UC Berkeley "The Beauty and Joy of Computing" : Functions (7)

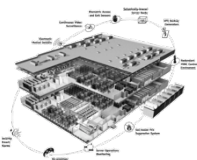
Types of input (there are more)

Sentences	<ul style="list-style-type: none"> Words separated by N spaces, $N \geq 0$ E.g., CS 10 is great
Word	<ul style="list-style-type: none"> Length ≥ 1, no spaces E.g., A, 3, #
Character	<ul style="list-style-type: none"> Length = 1 E.g., A, 3, #
Digit	<ul style="list-style-type: none"> 0-9 only E.g., 7

UC Berkeley "The Beauty and Joy of Computing" : Functions (8)

Why functions are great!



- If a function only depends on the information it gets as input, then nothing else can affect the output.
 - It can run on any computer and get the same answer.
- This makes it incredibly easy to parallelize functions.
 - Functional programming is a great model for writing software that runs on multiple systems at the same time.



Datacenter

UC Berkeley "The Beauty and Joy of Computing" : Functions (9)

Scratch → BYOB (Build Your Own Blocks)

- **Scratch**
 - Invented @ MIT
 - Maintained by MIT
 - Huge community
 - Sharing via Website
 - No functions ☹️
 - Scratch 2.0 in Flash
 - No iOS devices. ☹️
 - scratch.mit.edu
- **BYOB (and Snap!)**
 - Based on Scratch code
 - Maintained by Jens & Cal
 - Growing community
 - No sharing (yet) ☹️
 - Functions! 😊 ... "Blocks"
 - Snap! Is in HTML5
 - All devices ☺️
 - snap.berkeley.edu/run

UC Berkeley "The Beauty and Joy of Computing" : Functions (10)

Why use functions? (1)

pen down

repeat 4

 move 25 steps

 turn 90 degrees

pen up

pen down

repeat 4

 move 100 steps

 turn 90 degrees

pen up

pen down

repeat 4

 move 375 steps

 turn 90 degrees

pen up

Draw Square of Side 'length'

pen down

repeat 4

 move length steps

 turn 90 degrees

pen up

The power of generalization!

UC Berkeley "The Beauty and Joy of Computing" : Functions (11)

Why use functions? (2)

They can be composed together to make even more magnificent things.

They are literally the building blocks of almost everything that we create when we program.

We call the process of breaking big problems down into smaller tasks **functional decomposition**

join I am

join my age - your age years older than you.

UC Berkeley "The Beauty and Joy of Computing" : Functions (12)

Types of Blocks

- **Command**
 - No outputs, meant for side-effects
 - Not a function...

play drum 48 for 0.2 beats

move 10 steps
- **Reporter (Function)**
 - Any type of output

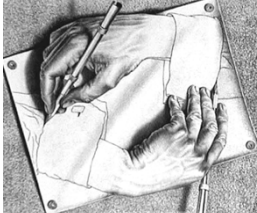
join hello world
- **Predicate (Function)**
 - Boolean output
 - (true or false)

and

UC Berkeley "The Beauty and Joy of Computing" : Functions (13)

Quick Preview: Recursion

Recursion is a technique for defining functions that use themselves to complete their own definition.



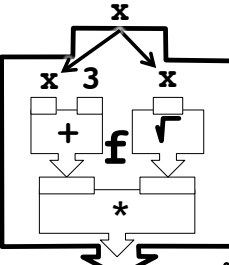
M. C. Escher · Drawing Hands

We will spend a lot of time on this.

UC Berkeley "The Beauty and Joy of Computing" : Functions (14)

Functions Summary

- **Computation is the evaluation of functions** $f(x) = (x+3) * \sqrt{x}$
 - Plugging pipes together
 - Each pipe, or function, has exactly 1 output
 - Functions can be input!
- **Features**
 - No state
 - E.g., variable assignments
 - No mutation
 - E.g., changing variable values
 - No side effects
- **Need BYOB/Snap!, and not Scratch 1.x**



UC Berkeley "The Beauty and Joy of Computing" : Functions (15)