

Compsci 101

Introduction

Part 1 of 3

Susan Rodger
January 26, 2021

```
var1 = 17
var2 = var1 + 12
var1 = "hello"
var2 = var1 * 3
```

B is for ...



- **Bug**
 - What you will always have and need to fix
- **Bits**
 - Zeros and Ones that are our C,G,A,T
- **Break**
 - An easy way out of a loop
- **Boolean**
 - Type that's true or false

Plan for the Day (PFTD)

- Names, types, and values in Python
- How to run a Python program
- Functions in Python

Exploring Language

- Do you first learn how to say something useful or do you learn the rules of grammar
 - Duolingo: El hombre bebe
 - How do you say the man thinks? The woman?
- Connecting to things you already know helps with learning!
- We'll do some bottom up exploration so we can solve problems
 - Variables - names, types, values

Names, Types, and Values

- What type is each of these files?
 - homework.pdf, blurp.mp4, egal.jpg, zoo.wav
- Does the name/suffix define the type or is it the bits in the file that defines the type?
 - *Value* of blurp.mp4 not the same as moo.mp4
 - *Type* of blurp.mp4 is the same as moo.mp4
 - *Name* of stairwaytoheaven.mp3 means ...

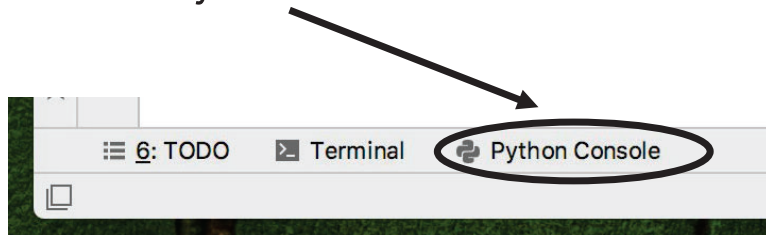


Numeric Python Building Blocks

- Numbers are not everything! But good start
 - Values and arithmetic expressions
 - Integer aka int: 0, 3, -2, 5, ...
 - Float: 2.5, 3.6673, 1.938e+120
 - Operators: +, -, *, /, **
 - Operators: // and %
- Demo in Python Console

Interactive Console

- Look in the bottom left corner of PyCharm
- Click on “Python Console”



Summary of Numbers

- Integers are arbitrarily large in Python 3
- Float values do not have infinite precision
 - Floats are for decimal values
- Be attentive to parentheses and precedence
- Understand / and // and %
 - Modulus or remainder

Python Strings

- A string is a sequence of characters
 - String literals use single or double quotes
 - "hello" and 'world' are both strings
- Operators we'll use: + and [:]
 - Concatenation and Slicing
 - Adding and taking apart?
 - *Today just adding*
- Demo in Console

Compsci 101 Introduction Part 2 of 3

Susan Rodger
January 26, 2021

```
var1 = 17
var2 = var1 + 12
var1 = "hello"
var2 = var1 * 3
```

Types and Conversion

- How do you convert a .mp4 to a .mov?
 - Change the bits from one format to another
- Can we add a string and an integer?
 - What does 5 + "cow" mean?
 - What does 5 * "cow" mean?
 - Why?

Using Python Console

- Not writing a whole program
- Just checking out values or writing simple code

- What is the difference in Python Console of:

```
>>> print("hello" + " " + "world")
```

hello world

```
>>> "hello" + " " + "world"
```

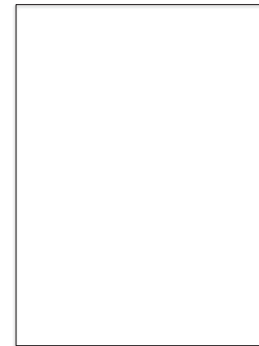
'hello world'

Variables

- We use variables to store values so we can use them and re-use them in expressions
 - Name associated with storage (spot in memory)
 - Assign value to a variable
- How to read: `num = 5`, `word = "hello"`
 - Why say 'gets' or 'is assigned' and not 'equals'
 - We'll use 'equals' later to mean equality

Variable idea

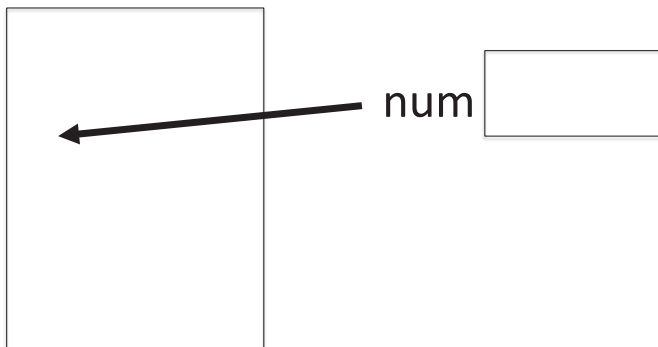
1) `num = 6`



Computer

Variable idea

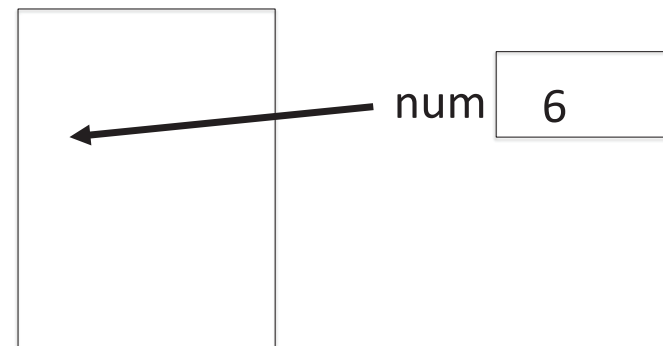
1) `num = 6`



Computer

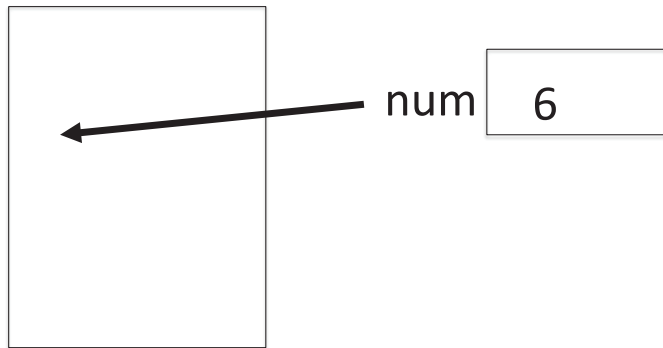
Variable idea

1) `num = 6`



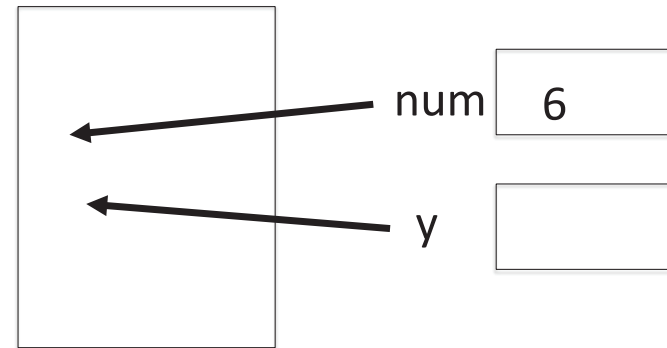
Computer

Variable idea
2) $y = \text{num} + 4$



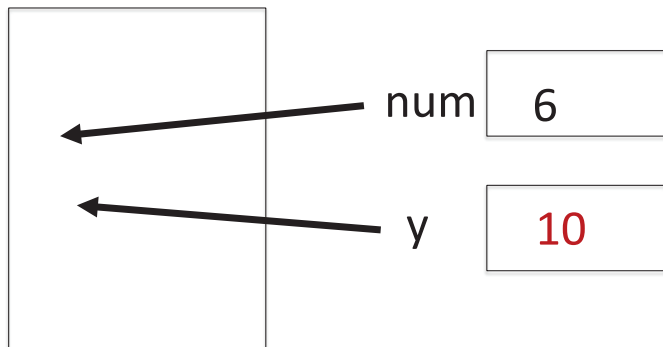
Computer

Variable idea
2) $y = \text{num} + 4$



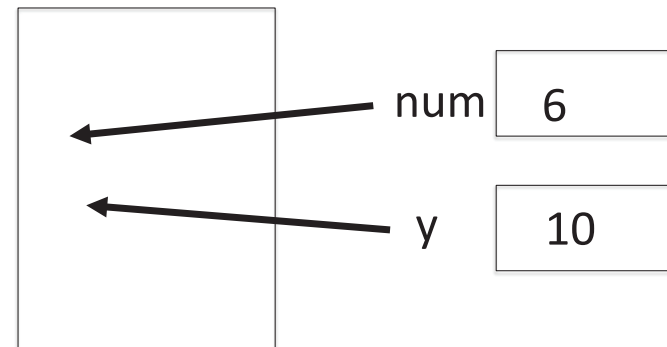
Computer

Variable idea
2) $y = \text{num} + 4$



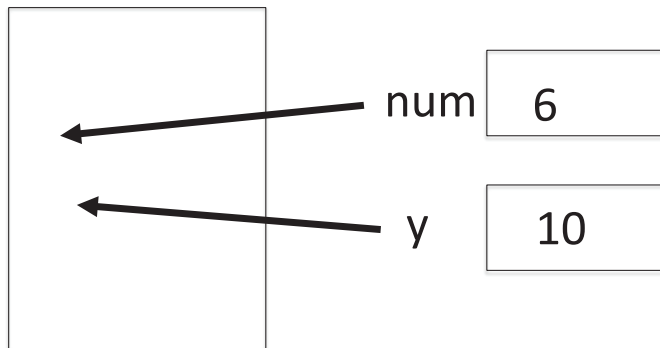
Computer

Variable idea
3) $\text{num} = y * 2$



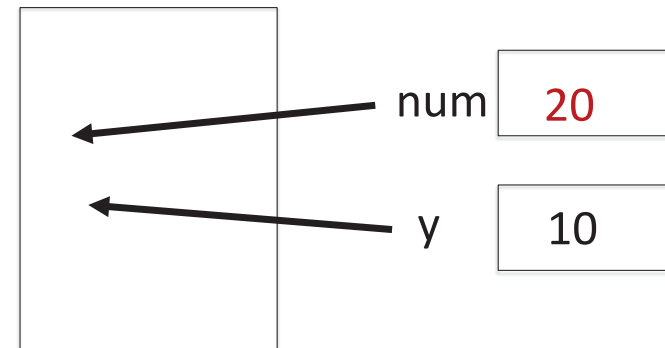
Computer

Variable idea
3) $\text{num} = y * 2$



Computer

Variable idea
3) $\text{num} = y * 2$



Computer

Anatomy of a variable

- Variables in Python have a type, changeable
 - Initially `var = 5`, change to `var = "hello"`
 - Use the `type(...)` function to determine type, but documentation/comments are better
- Variables are names/labels, references to an object stored elsewhere (basically)
 - My address is "202 Main Street"
 - That's the name/label, my house is elsewhere
 - For `var = "hello"`, the string is elsewhere

Subtleties

- Variables on LHS and RHS
 - Value compared to Name
 - LHS – Left Hand Side
 - RHS – Right Hand Side

```
num1 = 17
num2 = num1 + 12
```

- 1) Evaluate RHS, 2) Store in LHS

```
var1 = 17
var2 = var1 + 12
var1 = "hello"
var2 = var1 * 3
```

- What happens here?
 - Value compared to Name
- In expressions? What is the value

Compsci 101

Introduction

Part 3 of 3

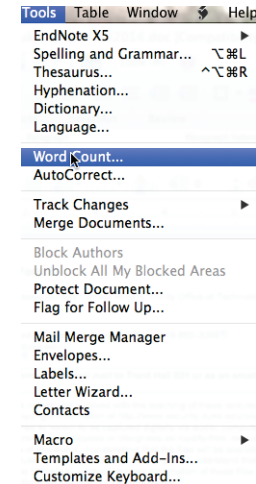
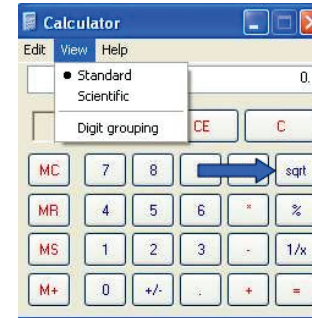
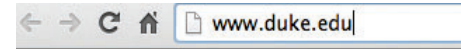
Susan Rodger
January 26, 2021

```
var1 = 17
var2 = var1 + 12
var1 = "hello"
var2 = var1 * 3
```

1/26/21

Compsci 101, Spring 2021 25

Functions in the Real World



1/26/21

Compsci 101, Spring 2021 26

Anatomy of a Function

- Named abstraction (over code chunk in Python)
 - Use the name to understand the conversion of input(s) to corresponding output
- Functions may have an input and always produce an output: `math.sqrt(25)`
 - Input is an argument
 - Variables holding inputs are parameters
 - Output is the return value
 - In Python call is name followed by parentheses

1/26/21

Compsci 101, Spring 2021 27

Simple Python Functions

- How many digits in `3**2500`?
 - How many characters in "hello world"
- **len** is a Python function
 - Input: sequence (e.g. a string)
 - Output: integer - length of sequence
- **type** is a function as are **int**, **float**, **str**
 - What if the input isn't part of the domain?

1/26/21

Compsci 101, Spring 2021 28

In Python version 3 print is a function

- **Functions have parentheses**
 - Arguments are provided in parentheses
 - We can `print(3+5)` or `print("hello")` or ...
 - What is returned by print?
- **When there is no return value...**
 - None is returned, it has no representation
 - Its type is `NoneType`
- **Note: in python version 2, print is NOT a function, looks different**