DS2000
Fall 2021
Handout: Variables and I/O

All programming languages must be able to:

1. remember things,
2. repeat things,
3. communicate, and
4. make decisions.

Variables are all about #1: remembering things. We have some value (a number, a word, a sentence, a list, etc.) that we want to remember, so we name it and stick it somewhere in the computer's memory. Now we can access that value again, or modify it, or use it in a calculation, and all we need to remember is the name we gave it.

Input/Output (I/O) is all about #3: communicating. We need data to crunch in data science -- we'll start by getting that from our program's end user. They communicate with our program by typing values (input) in the keyboard, and we save those values in variables. Once we've done some calculations with our variables, we tell the user what the result was (output).

Variables

We use variables when we want to **remember things.** We have some piece of data (a number, a word, a list, etc.) that we care about, so we give it a name and stick it somewhere in the computer's memory. Now we can access that value again, or modify it, or use it in a calculation, and all we need to remember is the name we gave it.

In Python, we always need to **initialize** a variable...

```
days = 3          # initialize a variable (for testing frequency!)
```

...and once it exists, we can modify it (same label, different value)

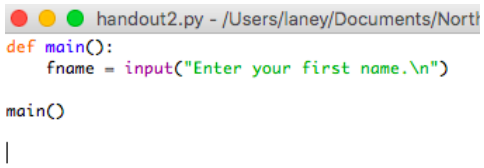
```
days = 5          # change the variable days from 3 to 5
```

In this example, the variable *days* is an **integer** (aka, a whole number). Python cares a lot about that -- it's the variable's **data type.** The data types we're mostly working with this semester are:

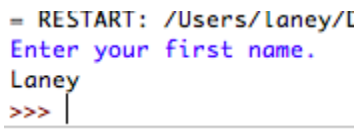
Data Type	Initialize/Modify Example
integer	days = 3
float	infection_rate = 0.06
boolean	result = False
string	patient = 'Laney Strange'

Input

We can also initialize/modify a variable by prompting the user for a value. Python has a built-in function called **`input`**. Python will wait patiently on the terminal until the user types something and hits enter.



Source code we write to prompt the user.



User sees the prompt and types in the terminal (they don't care about the source code).

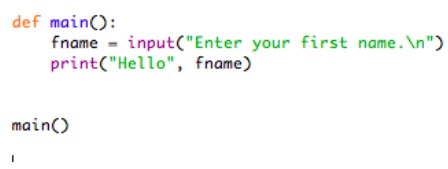
The **`input`** function always gives us a string. If you want to prompt the user for something other than a string, you need to do a little conversion:

Data Type	Input Example
integer	`x = int(input('Enter an int.\n'))`
float	`y = float(input('Enter a float.\n'))`
boolean	`b = bool(input('Enter a T/F value.\n'))`
string	`name = input('Enter your name.\n'))`

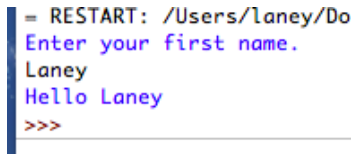
Output

In Python, output comes when we report something on the terminal -- usually the result of a calculation, or if we've gathered some information we want the user to see.

Python has a built-in function called **`print`**. We can give both literals and variables to this function and it puts them all together into one thing on the terminal.



Source code we write to print to the terminal.



User sees the output in the terminal (they don't care about the source code).

With ***print***, we need to put commas in between variables and literals. Python automatically puts a space after each one, and a linebreak at the end, but we can mess around with that too using ***sep*** and ***end***. Here are a couple more examples:

Example Source Code	Output
`x = 4` `y = 2.87`	`Variable x = 4 and y = 2.87`

`print('Variable x =', x, 'and y =', y)`	
`name = "Laney"` `print("Hello,", name, "!")`	`Hello, Laney !`
`name = "Laney"` `print("Hello, ", name, "!", sep = '')`	`Hello, Laney!`
`print('1 - ')` `print('collect underpants')`	`1 -` `collect underpants`
`print('1 - ', end = '')` `print('collect underpants')`	`1 - collect underpants`

Arithmetic Operators

Arithmetic Operations apply to numeric data types (float and int). As we just saw, arithmetic operations happen on the right-side of the assignment operator. The right-hand side is evaluated first, and then is assigned to whatever variable is on the left.

Python has your basic built-in arithmetic operators, plus a couple of fun ones.

Operator	**Operation**	**Example**	**Value of x**
+	Addition	x = 20 + 15	35
-	Subtraction	x = 20 - 15	5
*	Multiplication	x = 20 * 30	600
**	Exponent	x = 2 ** 4	16
/	Division	x = 20 / 15	1.3333
//	Integer Division	x = 20 // 15	1
%	Modulo	x = 20 / 15	5
+=	Increase	x = 1 x += 5	1 6
-=	Decrease	x = 1 x -= 1	1 0