

Getting Started With Python Programming: Part 1

- Tutorial: creating computer programs
- Variables
- Getting information from the user
- Common mathematical operators

Reminder!

- These course notes are mandatory
 - Get them before class and go over them before attending
- (If all else fails then look through them afterwards – at the very least to see what concepts/topics you are responsible for knowing).
 - It's the **first** step you should complete if you've missed lecture and need to catch up.
 - (The second step is to get the in class notes of a classmate).
 - After going through these notes the third step is to ask us for help in filling in any conceptual gaps.

James Tam

Tips For Success: Programming Sections

- (The previous 4 tips are still applicable but there's some tips specific to programming):
 - Take extensive notes: everything in class not just what the instructor "writes down" but also what he/she "says in class".
 - Some students may find when studying the lecture slides for the exam that they cannot understand concepts.
 - The extra "filling of the blanks" occurs during lecture so you need to annotate the slides with your own notes
 - After lectures have covered a particular concept/example
 - If you have time try writing the program on your own (without looking at the online examples or notes) in order to create a program that fulfills the same task as the example program
 - (It's one thing to see the solution to a problem explained, your depth of understanding will be deeper if you have to re-create it from scratch yourself).
 - JT's note: you may find this unnecessary for the simple examples in this section but it will be beneficial to do this when more complex concepts are covered (e.g. nested loops onwards)

James Tam

Python

- This is the name of the programming language that will be used to illustrate different programming concepts this semester:
 - My examples will be written in Python
 - Your assignments will be written in Python
- Some advantages (from Python dot org)
 - Free
 - Powerful
 - Widely used (Google, NASA, Yahoo, Electronic Arts, some Linux operating system scripts etc.)
- Named after a British comedy "Monty Python's Flying Circus"
 - Official website (Python the programming language, not the Monty Python comedy troop): <http://www.python.org>
 - An overview of the web site: <https://www.python.org/about/gettingstarted/>

James Tam

Python History

- Developed in the early 1990s by Guido van Rossum.
- Python was designed with a tradeoff in mind (from “*Python for everyone*” (Horstman and Necaise):
 - Pro: Python programmers could quickly *write programs* (and not be burdened with an overly difficult language)
 - Con: Python programs weren’t optimized to *run* as efficiently as programs written in some other languages.



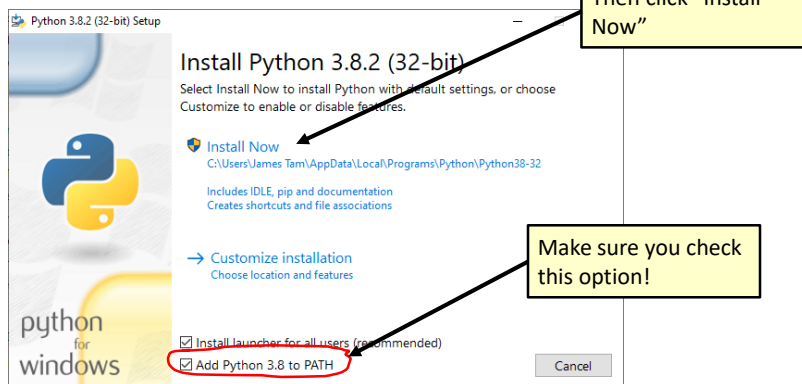
“Gawky and proud of it.”

From:
<http://www.python.org/~guido/>

James Tam

Working From Home (Installing Python)

- Getting Python (*get version 3.X* and not version 2.X)
 - <http://www.python.org/download/>



- Detailed information:
 - https://pages.cpsc.ucalgary.ca/~tamj/2021/217P/notes/pdf/installing_accessing_python.pdf

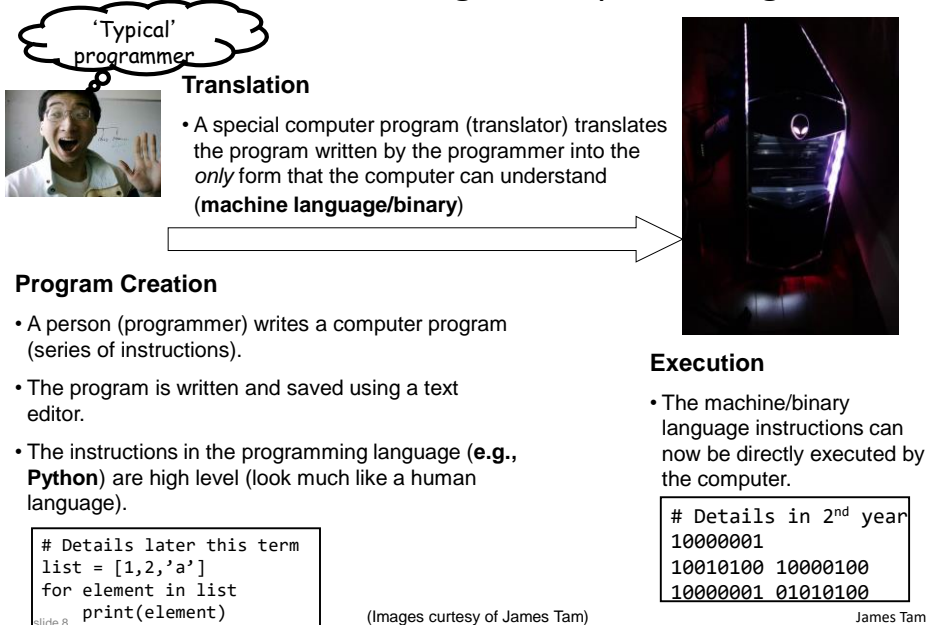
James Tam

Online Help: Official Python Site

- *Basic explanation* of concepts (for beginners: along with examples to illustrate)
 - <http://docs.python.org/py3k/tutorial/index.html>
 - (Skip the notes on the interactive mode for now where you type but don't save the program at the command line).
 - For this course you need to create a python program in a file and then run python on that file.

James Tam

The Process Of Creating A Computer Program



slide 8

James Tam

Location Of My Online Examples

- For this semester you can find them in D2L under 'Content'.
- Then look under the appropriately named folder which is listed by date and topic.
- Alternatively you can find them by looking under the "main grid" of the course website (look for the 'examples' link):
 - [https://pages.cpsc.ucalgary.ca/~tamj/2021/217P/#Main_grid: lecture & tutorial schedule, assignment information](https://pages.cpsc.ucalgary.ca/~tamj/2021/217P/#Main_grid:lecture&tutorial_schedule_assignment_information)

James Tam

The First Python Program

- Program name: small.py

Filename: 1small.py

```
print ("hello",end="")
```

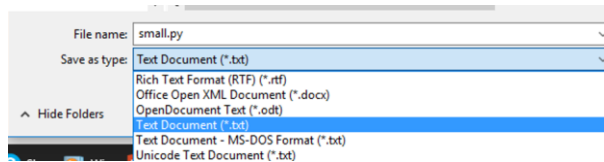
James Tam

Creating/Running Programs: Windows

• Step 1: Writing your program

—You need a text editor (e.g., *WordPad*, Notepad, Notepad++) to enter the program.

—It can be done using any editor that you want, but don't use a word processor (e.g., MS-Word) and remember to **save it as a text file** ending with the suffix dot-py “.py”



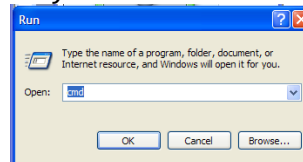
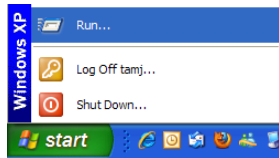
James Tam

Creating/Running Programs: One Operating System (2)

Step 2: Translating and running your program

— You need to open a command line to translate/run your Python program.

— The name of the Python translator is “Python”



— To translate/run your program type “python filename.py” at the command line.

- The first example program would be executed by typing “python 1small.py”
- For a program whose filename is called “output1.py” you would type “python output1.py”.

Output of program
(result of running the
program)

```
0:\231\examples\intro>python small.py
hello
```

Running
/translating
the program

James Tam

Python: Interactive Mode

- If you just type 'python' without the name of a file containing a python program then you will enter the "interactive mode" of python.
 - For now we won't be covering this mode

```

Select Command Prompt - python
Microsoft Windows [Version 10.0.18363.815]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\James Tam>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>

```

- To exit the mode either close the command line or at the command line type `exit()`

James Tam

Important Reminders

- Make sure you type the whole file name (including the part after the period) when you translate/run your program.
 - E.g., "python small.py"
- Unless you are very familiar with your operating system when you translate/run a program you should first navigate to the directory/folder where your Python program resides.
 - JT: the 'cd' command changes your directory (Windows and UNIX although something different is needed when changing Windows drives)
 - Suppose my program was under:
 - C:\231 (Windows)
 - OR
 - /home/231 (UNIX)
 - To reach this location you could (shortcuts excluded for now) then type:
 - cd c:\231 (Windows)
 - OR
 - cd /home/231 (UNIX)

James Tam

Section Summary: Writing A Small “Hello World” Program

- You should know exactly what is required to create/run a simple, executable Python program.
 - While you may not be able to create a new program from scratch at this point, you should be able to enter/run `small.py` yourself.

James Tam

Variables

- Set aside a location in memory.
- Used to store information (temporary).
 - This location can store one ‘piece’ of information.
 - Putting another piece of information at an existing location **overwrites** previous information.
 - *At most* the information will be accessible as long as the program runs i.e., it’s temporary
- Some types of information which can be stored in variables include: integer (whole), floating point (fractional), strings (essentially any characters you can type and more)

Format (creating):

`<name of variable> = <Information to be stored in the variable>`

Examples (creating):

- Integer (e.g., `num1 = 10`)
- Floating point (e.g., `num2 = 10.0`)
- Strings: alpha, numeric, other characters enclosed in quotes.
 - e.g., `name = "james"`
 - To be safe get in the habit of using double (and not single) quotes

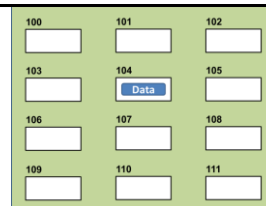


Image courtesy of James Tam

slide 16

James Tam

The Assignment Operator: =

- The assignment operator '=' used in writing computer programs does not have the same meaning as mathematics.
 - Don't mix them up!
- Example:
 - `y = 3` (what is stored in 'y' at this point)
 - `x = y` (what is stored in 'x', 'y' at this point)
 - `y = 6` (what is stored in 'x', 'y' at this point)
- What is the end result? How was this derived (what are the intermediate
- **Name of the full example:** 2assignment.py

```
3
3 3
3 6
```

- Quick tip: to show what a variable currently contains put the name of the variable without quotes inside the round brackets for the print function.

James Tam

Variable Naming Conventions

- Python requirements:
 - Rules built into the Python language for writing a program.
 - Somewhat analogous to the grammar of a 'human' language.
 - If the rules are violated then the typical outcome is the program cannot be translated (nor run).
 - A language such as Python may allow for a partial execution (it runs until the error is encountered).
- Style requirements:
 - Approaches for producing a well written program.
 - (The real life analogy is that something written in a human language may follow the grammar but still be poorly written).
 - If style requirements are not followed then the program can still be translated but there may be other problems (more on this during the term).

James Tam

Variable Naming Conventions (2)

1. Style requirement: The name should be meaningful.
2. Style and Python requirement: Names *must* start with a letter (Python requirement) and *should not* begin with an underscore (style requirement).
3. Style requirement: Names are case sensitive but avoid distinguishing variable names only by case.

Examples

#1:

age (yes) x, y (no)

#2

height (yes) 2x, _height (no)

#3

Name, name, nAme (no to this trio)

James Tam

Variable Naming Conventions (2)

4. Style requirement: Variable names should generally be all lower case (see next point for the exception).
5. Style requirement: For names composed of multiple words separate each word by capitalizing the first letter of each word (save for the first word) or by using an underscore. (Either approach is acceptable but don't mix and match.)
6. Python requirement: Can't be a keyword (see next slide).

Examples

#4:

age, height, weight (yes)
Age, HEIGHT (no)

#5

firstName, last_name
(yes to either approach)

James Tam

Key Words In Python¹

and	del	from	not	while
as	elif	global	or	with
assert	else	if	pass	yield
break	except	import	print	
class	exec	in	raise	
continue	finally	is	return	
def	for	lambda	try	

¹ From "Starting out with Python" by Tony Gaddis

James Tam

Variable Naming Conventions: Bottom Line

- Both Python and style requirements should be followed when creating your variables.

James Tam

Extra Practice

- Come up with example names that violate and conform to the naming conventions.
 - (You will have to go through this process as you write your programs anyhow so it's a good idea to take about 5 – 10 minutes to make sure that you know the requirements).

James Tam

Section Summary: Variables

- What is a variable
- What are some types of variables available in Python
- How to create a variable in Python
- What are naming conventions for variables

James Tam

Displaying Output Using The Print() Function

- This function takes zero or more arguments (inputs)
 - Multiple arguments are separated with commas
 - `print()` will display all the arguments followed by a blank line (move the cursor down a line).
 - `end=""` isn't mandatory but can be useful to prevent Python from adding the extra line (when precise formatting is needed)
 - Zero arguments just displays a blank line
- **Name of the full example:** 3output.py

```
print("hi")
```



```
hi
```

```
print("hey",end="")
```



```
hey-sup?
```

```
print("-sup?")
```

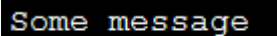
James Tam

Print("... ") Vs. Print(<name>)

- Enclosing the value in brackets with quotes means the value in between the quotes will be literally displayed onscreen.
- Excluding the quotes will display the contents of a memory location.
- **Name of the full example:** 4output.py

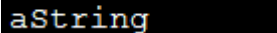
```
aString = "Some message"
```

```
print(aString)
```



```
Some message
```

```
print("aString")
```



```
aString
```

James Tam

- Used to format text output (free form and to reduce the number of calls to the `print()` function)
- The way in which the text is typed into the program is exactly the way in which the text will appear onscreen.
- Name of the full example:** 6formatting.py

```
print("""
┌───┐ ┌───┐ ┌───┐ ┌───┐
│   │ │   │ │   │ │   │
│   │ │   │ │   │ │   │
│   │ │   │ │   │ │   │
└───┘ └───┘ └───┘ └───┘
""")
```

```
*****
* Middle Earth: The Mines of Moria *
*****

This game allows you replay a portion of JRR Tolkien's
trilogy (TM). You control the fate of the Fellowship of the
Ring as they navigate the dark and perilous Mines of Moria
which leads to the ancient Dwarven city of Khazad-dum. Beware!
Numerous orc companies prowl the underdark and the demonic
Balrog will seek thee out. Run!, don't walk to the Misty
Mountains and begin your epic quest today.

This game has been created for education purposes only and is
not meant as a challenge to the copyright licenses of either
Tolkien Enterprises or New Line Entertainment

<Hit return/enter to continue>
```

James Tam

Side Note, Output: Python 3 Vs. Python 2

- Python 3: to be consistent all functions require brackets to enclose the arguments/function inputs
 - E.g. `print("Sup?")`
- Python 2 does not explicitly bracket all functions
 - E.g. `print "Sup?"`
- There are other differences so make sure you are writing programs that follow the Python 3 and not Python 2 syntax.

James Tam

Arithmetic Operators

- Name of the full example: `7operators.py`

Operator	Description	Example
=	Assignment	<code>num = 7</code>
+	Addition	<code>num = 2 + 2</code>
-	Subtraction	<code>num = 6 - 4</code>
*	Multiplication	<code>num = 5 * 4</code>
/	Division	<code>num = 9 / 2</code> 4.5
//	Integer division	<code>num = 9 // 2</code> 4
%	Modulo	<code>num = 9 % 2</code> 1
**	Exponent	<code>num = 9 ** 2</code> 81

James Tam

Order Of Operation

- First level of precedence: top to bottom
- Second level of precedence
 - If there are multiple operations that are on the same level then precedence goes from left to right.

()	Brackets (inner before outer)
**	Exponent
*, /, //, %	Multiplication, division, modulo
+, -	Addition, subtraction
=	Assignment

Example

num = 3 * 2 ** 3

Vs.

num = (3 * 2) ** 3

James Tam

Order Of Operation And Style

- Even for languages where there are clear rules of precedence (e.g., Java, Python) it's good style to explicitly bracket your operations and use blank spaces as separators.


```
x = (a * b) + (c / d)
```
- It not only makes it easier to read complex formulas but also a good habit for languages where precedence is not always clear (e.g., C++, C).

James Tam

After This Section You Should Now Know

- How to create, translate and run Python programs.
- Variables:
 - What they are used for
 - How to access and change the value of a variable
 - Conventions for naming variables
 - How information is stored differently with different types of variables, converting between types
- Output:
 - How to display messages that are a constant string or the value stored in a memory location (variable or constant) onscreen with `print()`
- How/why use triple quoted output
- What are the Python operators for common mathematical operations
- How do the precedence rules/order of operation work in Python

James Tam