Getting Started With Python Programming: Part I

- 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 example programs 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.)
- Some starting python resources
  - Official website: 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.
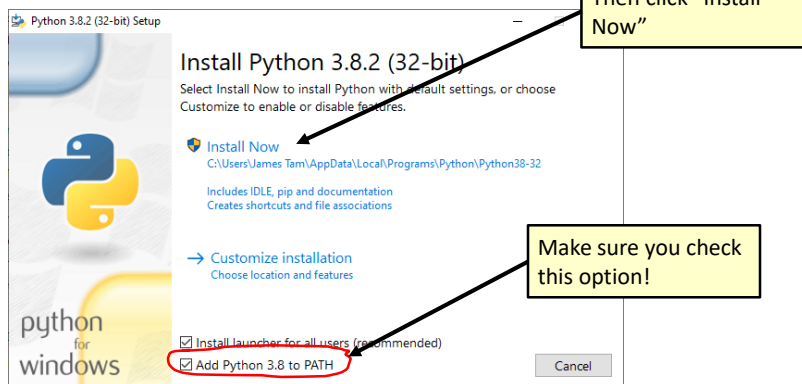
"Gawky and proud of it."

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

James Tam

Working From Home (Installing Python): Windows

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

Then click "Install Now"

Python 3.8.2 (32-bit) Setup

Install Python 3.8.2 (32-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

→ **Install Now**
C:\Users\James Tam\AppData\Local\Programs\Python\Python38-32

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ **Customize installation**
Choose location and features

Make sure you check this option!

☑ Install launcher for all users (recommended)
☑ Add Python 3.8 to PATH   Cancel

python for windows

  - 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 – it's where you don't save the program which leaves you nothing to submit for assignments).
  - For this course you need to <u>create a python program in a file</u> and then run the program defined in the file.

James Tam

Creating A Computer Program

1. A programmer writes the instructions of the program in high level (human can read and understand) language.

    - Examples: C, C++, java, python

    ```
    # Details later this term
    list = [1,2,'a']
    for element in list
        print(element)
    ```

2. The program must be created and saved using a text editor (e.g. Notepad, WordPad or the editor that comes with python IDLE).

    - Don't use a word processor!

3. The program is then translated into binary/machine language (the only form that the computer can understand).

    ```
    # Details in 2nd year
    10000001
    10010100 10000100
    10000001 01010100
    ```

James Tam

Location Of My Online Examples

- For this semester you can find them in D2L under: `Content->Lectures`
- 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/2022/217P/#Main_grid:_lecture_&_tutorial_schedule,_assignment_information

## The First Python Program

- Program name: small.py

Filename: `1small.py`

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

Creating/Running Programs: Windows

- **Step 0: Before writing your program start IDLE (python editor)**
  - The editor automatically comes installed with your python download so it's the one that we will officially support.
  - You can use other editors but keep in mind **you may be on your own if you have problems with that editor**!
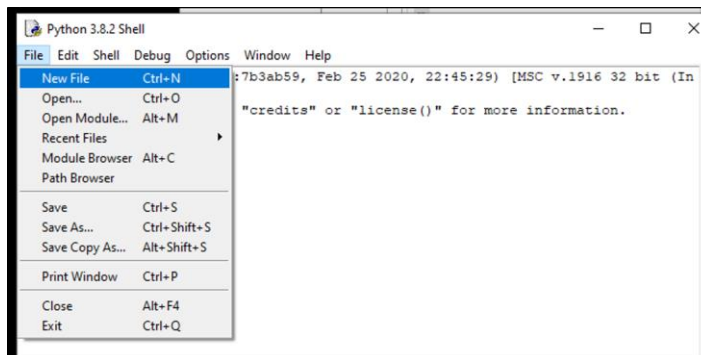   - Starting IDLE (Windows)
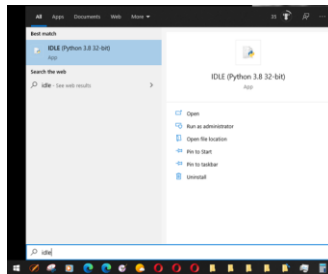
  –One Apple resource for using IDLE:
  https://www.python.org/download/mac/tcltk/

James Tam

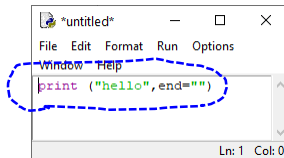# **Creating**/Running Programs: Windows (2)

- **Step 1**: Starting the python program editor from IDLE: `File->New File`

James Tam

Creating/Running Programs: Windows (3)

- **Step 2**: Type your program into the editor window that appears (appears if you completed the previous step correctly):
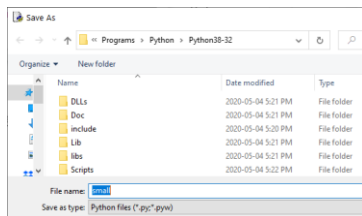

- (The specifics of <u>what</u> to type into the editor will form the bulk of teaching material for this semester – details are coming soon).
  - For now you can just type in what you see in the above image.

James Tam

Creating/Running Programs: Windows (4)

- **Step 3**: Name your program: `File -> Save`

  - Using the IDLE editor you can simply save under the default file type.
    - (If you use another text editor such as WordPad then you will have to save it as a text file <u>and</u> make sure the filename ends with the right suffix ".py").
    - Save the program in a easy to remember location (don't use the default location – it's the install location of python and not easy to remember).
  - **Standards for naming programs**
    - Only use alphabetic or numeric characters in the save name
    - Don't include spaces or any other characters!

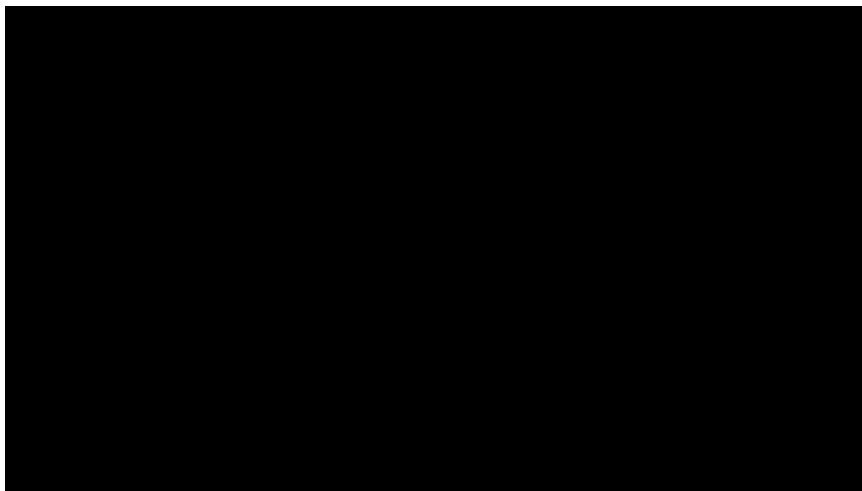James Tam

Creating/**Running** Programs: Windows (5)

- **Don't** try to open or run your program by clicking on it!
  - By default Windows will try to run the program in a new popup window and then close the window after the program finishes.
  - Typical result: the appears and then almost immediately closes.
  - Because the default behavior of files containing python programs is to execute the program the university MS-Exchange email server will block such file attachments.
    - Workaround for mailing a python program: rename the file name from `.py` to `.txt` before attaching it.
      - How to rename a file in Windows 10:
        - » https://www.howtogeek.com/665514/6-ways-to-rename-files-and-folders-in-windows-10/
      - (How to view filename extensions in Windows 10):
        - » https://www.howtogeek.com/205086/beginner-how-to-make-windows-show-file-extensions/
      - A MAC resource:
        - » https://support.apple.com/en-ca/guide/mac-help/mchlp2304/mac

Video: showing how useless it is to just click on a file containing a python program (use the method that will be covered shortly):
https://pages.cpsc.ucalgary.ca/~tamj/resources/python/Clicking_on_a_python_program_annotated.mp4

James Tam

Embedded Video (In Case You Can't See It Via The Link On The Last Screen).

James Tam

Requirements For Naming The File Containing Your Program

- As mentioned: no spaces!
  - Okay: `A1.py`, `gradeProgram.py`
  - Not Okay: grade program.py
- Just stick to using alphabetic characters, numbers may be used but the first character should only be alphabetic.
  - Okay: `A1Tam.py`, `A1Sept12.py`
  - Not Okay: `A1(1).py`, `9-25-2015.py`

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.
- You should also become familiar with requirements for naming the file which contains your python program.

James Tam

Variables

100	101	102
103	104 Data	105
106	107	108
109	110	111

- Set aside a location in memory.
- Used to store information (temporary).

Image curtesy of James Tam

  - 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 number storage only),
  - floating point (fractional),
  - strings (character information - essentially any characters you can type and more)
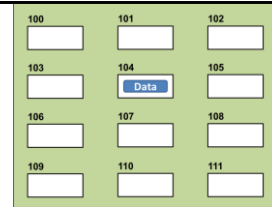
slide 19   James Tam

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 after using the assignment operator: to show what a variable currently contains put the name of the variable <u>without quotes</u> inside the round brackets for the print function e.g.,
    - num = 888
    - print(num)

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 (not the least of which is a reduced assignment grade - 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

Some Key Words In Python

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

Program instructions

```
print("hi")
print = "bye"
print("hi again")
```

Result of running the program

```
hi
Traceback (most recent call last):
  File "F:/work home/217F 2021/examples/keyword.py", line 3, in <module>
    print("hi again")
TypeError: 'str' object is not callable
```

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

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 (the cursor is automatically moved to the next 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?")
```
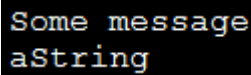
Print("… ") Vs. Print(<*name*>)

- 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)
print("aString")
```

```
Some message
aString
```
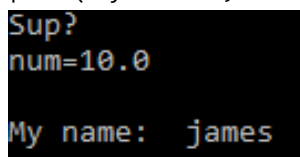
Print("… ") Vs. Print(<name>): 2

Format:
```
print(arg1,arg2 … )1
```
Name of the full example: 5output.py
```
num = 10.0
name = "james"
print("Sup?")
print("num=", end="")
print(num)
print()
print("My name: ", name)
```

```
Sup?
num=10.0

My name:  james
```

Student exercise 1: remove these quotes and see if you can correctly predict the results.

Student exercise 2: remove parts:
1) end=""
2) 2) print()

See if you can correctly predict the results.

1 From what you've learned thus far each argument can be a constant string or name of a variable.

Triple Quoted Output

- 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 ("""
 / ---|    /          /  |/ |
| |      / /|      / /|    /| | | |
| |         / -    / / \/     | | |
| |  -   / /   / /       | | |
|  -| / /    | | / /       | | |
\____|
""")
```
From Python Programming (2nd Edition) by Michael Dawson

```
***********************************
* 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 proposes only and is
not meant as a challenge to the copywrite licenses of either
Tolkien Enterprises or New Line Entertainment

<Hit return/enter to continue>
```
From a CPSC 231 assignment (image courtesy of James Tam)

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 syntax** and not Python 2 syntax.

James Tam

Arithmetic **Operators**

• **Name of the full example:** `7operators.py`

Operator	Description	Example	
=	Assignment	num **=** 7	
+	Addition	num = 2 **+** 2	
-	Subtraction	num = 6 **-** 4	
*	Multiplication	num = 5 ***** 4	
/	Division (real number)	num = 9 **/** 2	4.5
//	Integer division	num = 9 **//** 2	4
%	Modulo (remainder)	num = 9 **%** 2	1
**	Exponent	num = 9 ****** 2	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.

1st	( )	Brackets (inner before outer)
2nd	**	Exponent
3rd	*, /, //, %	Multiplication, division, modulo
4th	+, -	Addition, subtraction
5th	=	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