APS106

UNIVERSITY OF TORONTO

Tutorial 1 - Week 2

We'll be starting at the 10 minute mark

if nothing else, write #cleancode

Agenda

1. TA Introductions
2. Logistics
3. Introduction to Wing 101 IDE
4. Programming concepts: types, variables, operators, expressions, errors
5. Practice problems
6. Questions?

Introduction - TA

Ali Howidi

(TUT# + TUT#)

Current studies: Second year MASc BME student

Research/other interests: Signal analysis and machine learning for biosignals

Tutorial Logistics

- Tutorials are for your benefit - no grading
  - We will review previous weeks labs & lecture content
- Be sure to ask lots of questions and have Python open. We are here to help you!
- Questions outside of tutorial time?
  - Post to Piazza - all TAs/instructors and your peers can answer questions quickly
  - Coffee Time – drop-in hours for 1on1 help

Coffee Time With TAs!

- These are hours for any extra help you need for the course or programming in general ☺
  - also known as **office hours**

Please go to the following link to submit your response.

https://forms.gle/4NeZXciXTnUBZFTQ7

Virutal Tutorial and Virutal Labs! !

Please go to the following links to submit your response.

Tutorial:

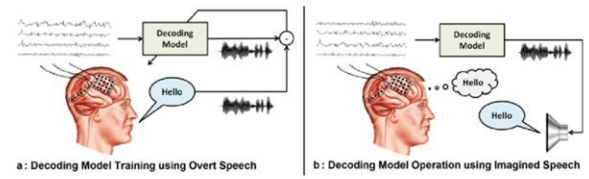
https://forms.gle/CPC5waK4m4wJgQnp9

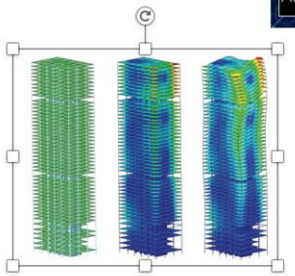
Labs:

https://forms.gle/2GrSNK7BoNPafjYr7

Why code?[1]

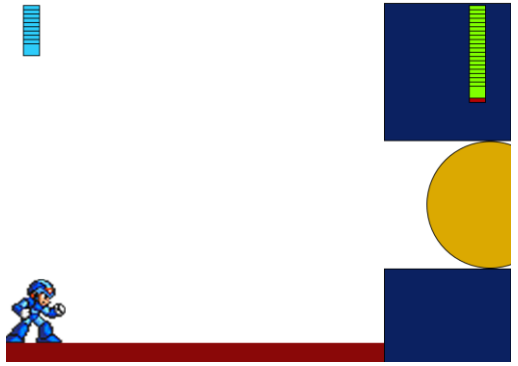
- Programming is awesome!
- Computers are everywhere, programming is a boundless opportunity
- Software engineering can be applied to almost any context in the world

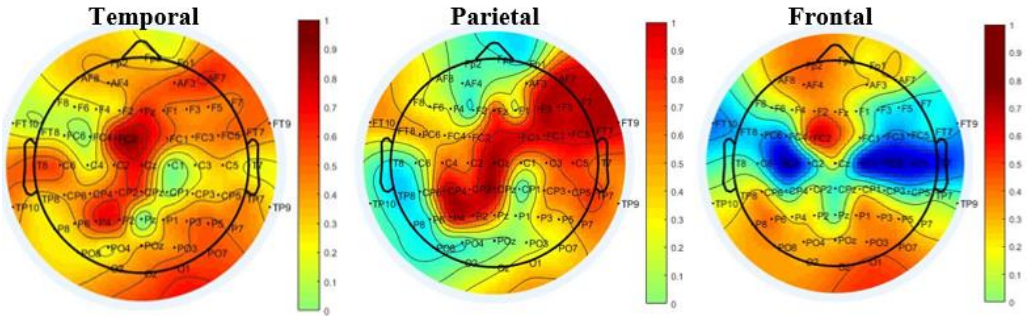


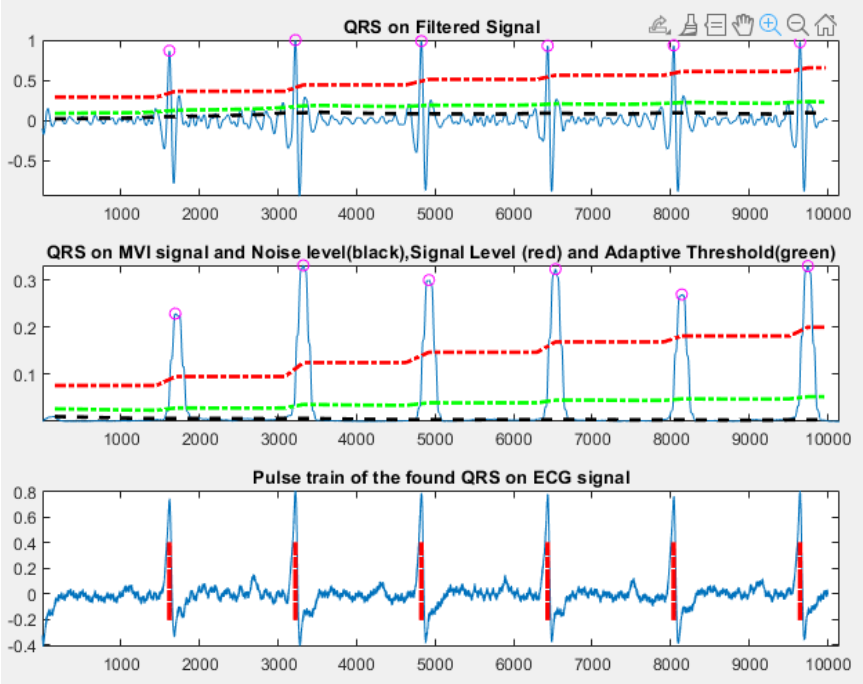


Why code?[2]

- Regardless of background, coding is always an achievable skill
- Software engineering can be applied to almost any context in the world
- **NEVER THINK THAT YOU "CAN'T UNDERSTAND" CODING**







What would you want to ideally learn how to do with computers?

slido

What would you want to ideally learn how to do with computers?

ⓘ Start presenting to display the poll results on this slide.

Coding experience?

slido

How would you rate your coding experience

ⓘ Start presenting to display the poll results on this slide.

APS106

UNIVERSITY OF TORONTO

Intro to Wing 101 IDE

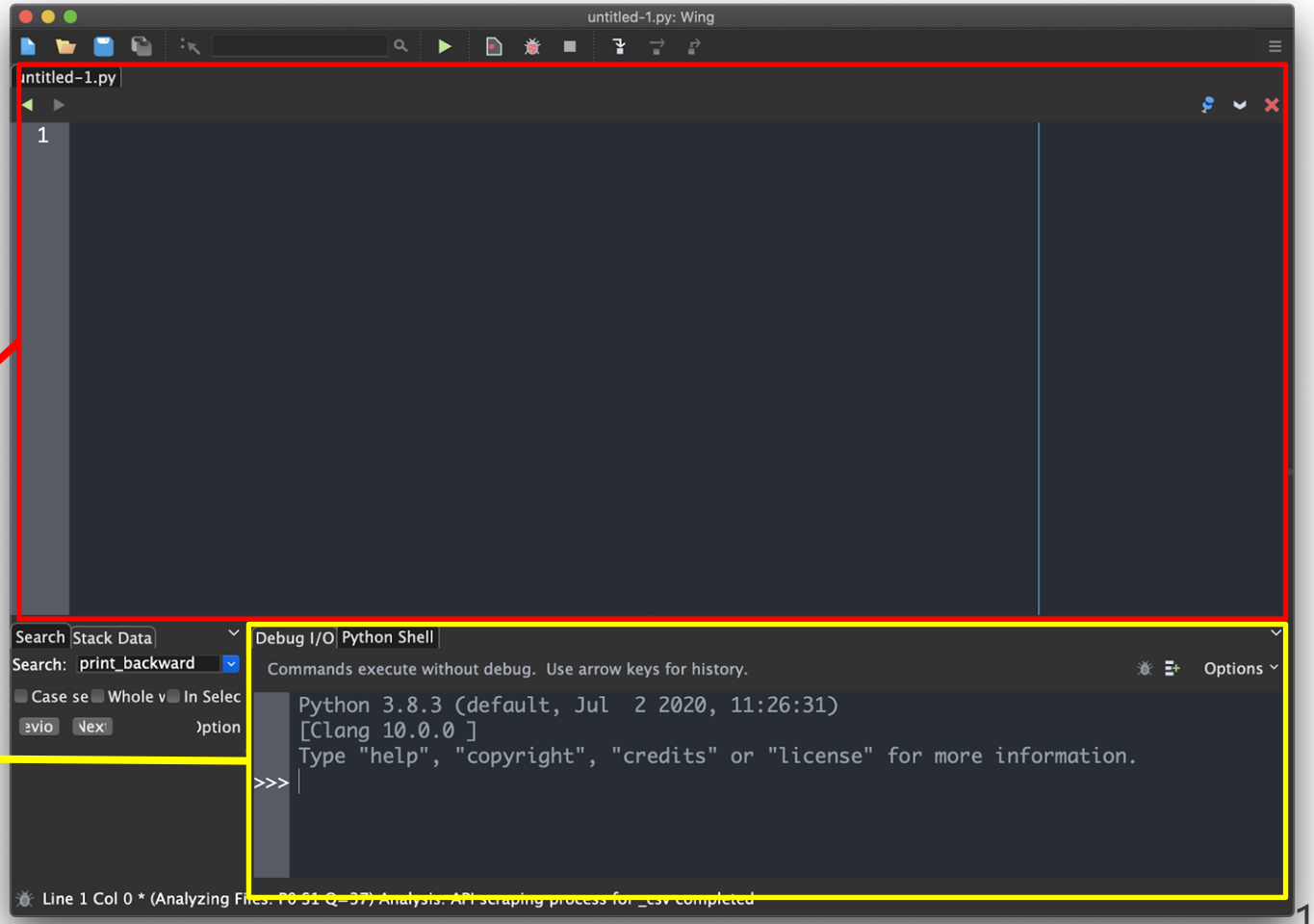
if nothing else, write #cleancode

Wing101

Wing 101 is an Integrated Development Environment

Script/program editing window

Python Shell



Click on the green chevron to run your code

Script Name

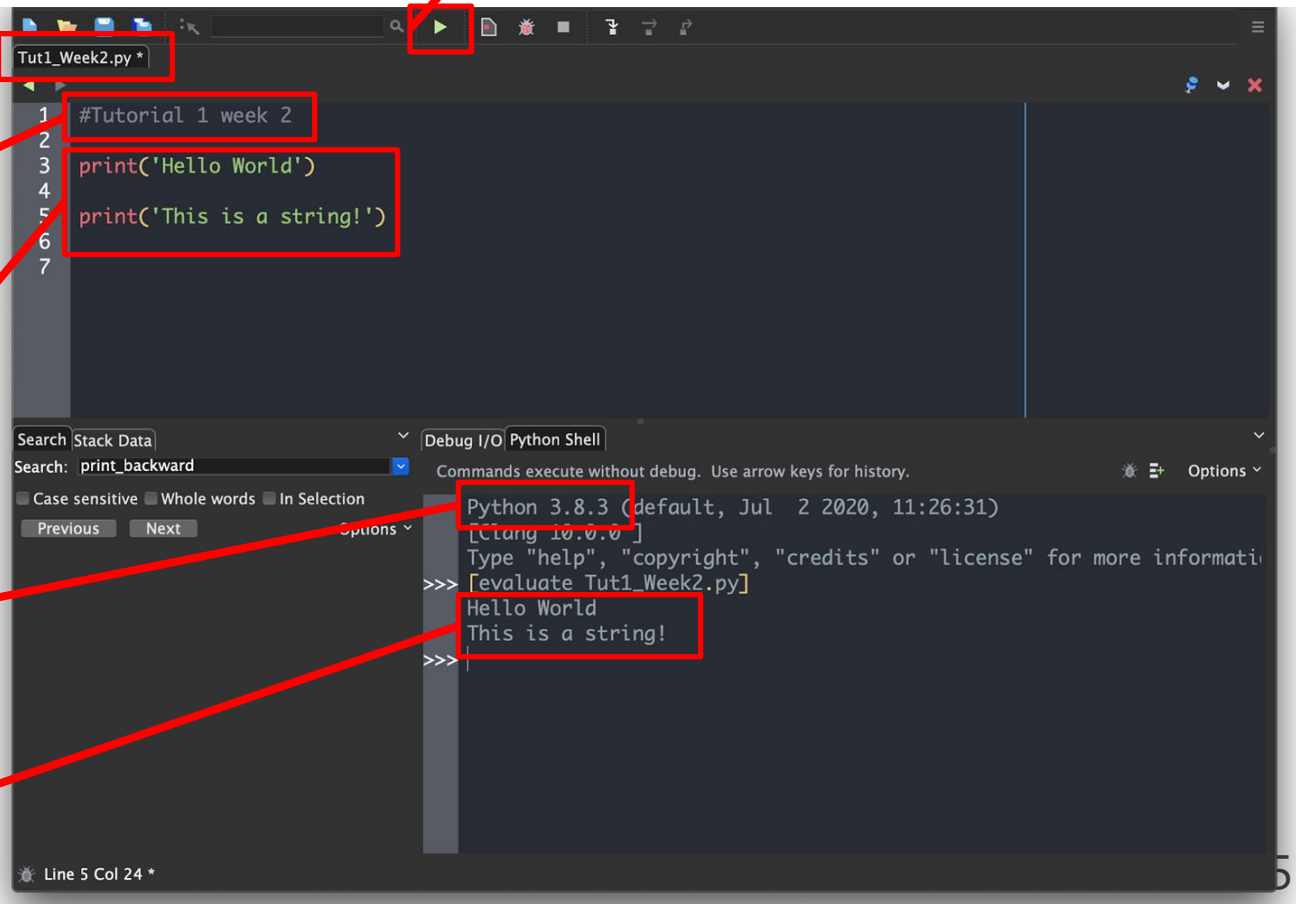
Single-line comments. They start with a “#”

Tip: Use these to make your script easier to follow

Statements that will be executed

Python version in use. Make sure you are using Python3 !

Python code can be written and run directly in the shell



APS106

UNIVERSITY OF TORONTO

Programming Concepts

Values, Types, and Variables

if nothing else, write #cleancode

Values and Types in Programming

Value: the representation of some entity that can be manipulated by a program, e.g., 5.

Type: a set of values and the operations that can be performed on those values

Examples of Python types and values:

- **int:** the set of integer numbers
  - sample value:

```
>>> 5
```

- **float:** the set of (double-precision) floating-point numbers
  - sample value:

```
>>> 1.55666
```

- **str:** the set of sequences of characters
  - sample values:

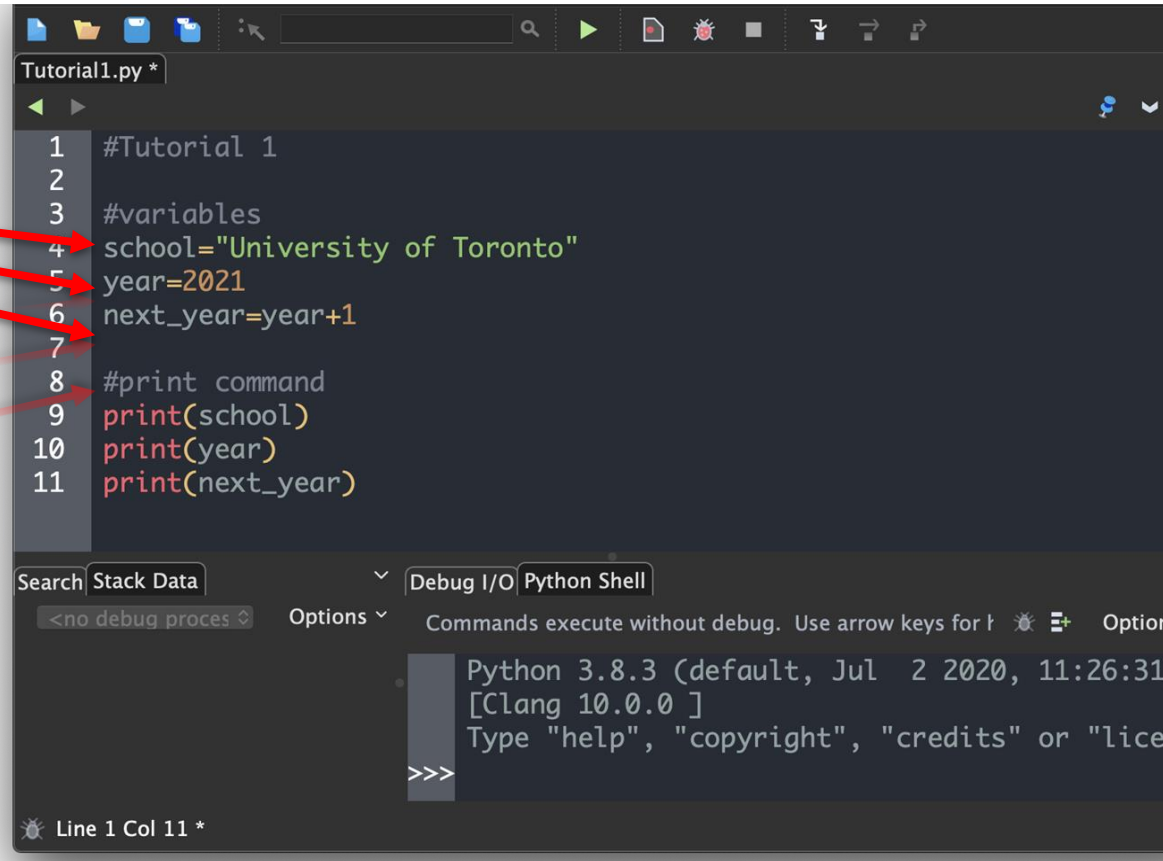
```
>>> "This is a string!"
```

```
>>> "12345"
```

Variables

- A **variable** is a "container" for a value.
- Variables are associated with values via **assignment statements.**
  - *assignment statements* in Python:

```
variable_name = expression
```



In Python:

- **variable names must:**
  - start with a letter or _
  - contain only letters, digits or _
- **variables have types** (a variable has the type of the value assigned to it)

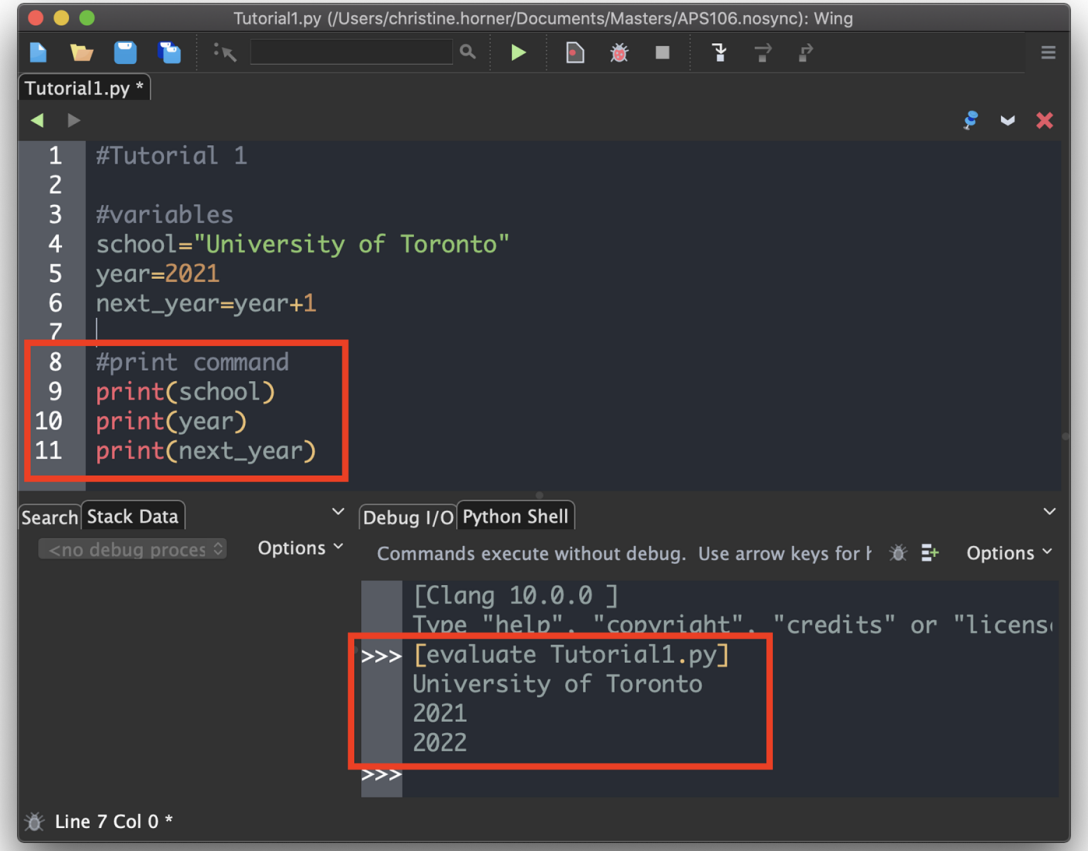
Tip: use meaningful variable names !

Python Function `print()`

In Python, we can use the built-in function **`print()`** to display objects/values in the shell.

In particular, we can use **`print()`** to display the value of program variables.

TO DO: run this code with and without the `print` statements.



APS106

UNIVERSITY OF TORONTO

Programming Concepts

Operators and Expressions

if nothing else, write #cleancode

Operators - Basics

- Operators are programming constructs that generally behave like functions.
- An operator is, generally, represented by a symbol, e.g.: `+`, `/`, `-`, `//`, `%`, `<`, `==`, `>`, `and`

Examples of Python operators

Type	Name	Symbol	Operand Data Type(s)	Arity
arithmetic	integer division	//	Number	binary
arithmetic	modulo	%	Number	binary
relational	less than	<	various types	binary
logical	logical conjunction	and	all data types	binary

Do not worry about these 2 for now

```
>>> 5//2
2
```

```
>>> 5%2
1
```

```
>>> 1<2 and 2<1
False
```

```
>>> 1<2
True
```

Expressions

- An ***expression*** is a combination of values (literals), variables, operators, and parentheses.
- An ***arithmetic expression*** is an expression that contains numerical values, arithmetic operators and parentheses.
  - The operators in an **un-parenthesized** arithmetic expression are evaluated in **order of precedence.**

Python Operator (symbol)	Name (Operation)
+	addition
-	subtraction
*	multiplication
**	exponentiation
/	division
//	integer division
%	modulo (remainder)

Python Examples of Simple Arithmetic Expressions	English description	Result
11 + 3	11 plus 3	14
5 - 19	5 minus 19	-14
7 * 4	7 multiplied by 4	28
2 ** 5	2 to the power of 5	32
9 / 2	9 divided by 2	4.5
9 // 2	9 divided by 2	4
9 % 2	9 mod 2	1

Arithmetic Expressions

Arithmetic expressions follow the same rules learned in math class!

The operators in an un-parenthesized expression are evaluated in order of precedence.

Arithmetic Operations

Operator	Operation	Expression	English description	Result
`+`	addition	`11 + 3`	11 plus 3	`14`
`-`	subtraction	`5 - 19`	5 minus 19	`-14`
`*`	multiplication	`7 * 4`	7 multiplied by 4	`28`
`**`	exponentiation	`2 ** 5`	2 to the power of 5	`32`
`/`	division	`9 / 2`	9 divided by 2	`4.5`
`//`	integer division	`9 // 2`	9 divided by 2	`4`
`%`	modulo (remainder)	`9 % 2`	9 mod 2	`1`

Gries, Table 1, p. 13

Operator	Precedence
**	highest
- (negation)	
*, /, //, %	
+, -	lowest

Gries, Table 2, p. 15

Operator Precedence

Operator precedence, or the **order of operations**, is a collection of rules about which actions to perform first in order to evaluate a given expression.

- For example, in all programming languages, the arithmetic operators, e.g., multiplication, division, integer division, modulo, addition, subtraction, are evaluated according to the order of precedence of their corresponding math operators.
  - **Precedence rules for other operators, e.g., relational operators, are language-dependent !**

Operator	Precedence
**	highest
- (negation)	
*, /, //, %	
+, -	lowest

```
>>> 100*(50**(2+3))-5
31249999995
```

Give it a try! Arithmetic Operation Examples

Integer division: //

```
>>> 5//2
2
```

Modulo: %

```
>>> 5%2
1
```

Arithmetic Precedence

```
>>> 100*(50**(2+3))-5
31249999995
```

Augmented Operators in Python

Operator	Expression	Identical Expression
+=	x = 7 x += 2	x = 7 x = x + 2
-=	x = 7 x -= 2	x = 7 x = x - 2
*=	x = 7 x *= 2	x = 7 x = x * 2
/=	x = 7 x /= 2	x = 7 x = x / 2
//=	x = 7 x //= 2	x = 7 x = x // 2
%=	x = 7 x %= 2	x = 7 x = x % 2
**=	x = 7 x **= 2	x = 7 x = x ** 2

APS106

UNIVERSITY OF TORONTO

Programming Concepts

Errors

if nothing else, write #cleancode

Error Types[1]

Name	Definition	Timing	Python Examples

Syntax error	An error due to the code not being syntactically valid (i.e., not obeying the grammar of the programming language)	Usually discovered during compilation	`>>> (3 + 2 * 4`
			`Syntax Error: unmatched ')'`
			`Expression (3 + 2 * 4 contains an open parenthesis but not a corresponding close parenthesis`

Semantic error	An error due to the code not following the semantic rules of a programming language, e.g., when operators are used improperly or when an attempt is made to use a variable that has not yet been associated with a value, or	Usually discovered during compilation	`>>> "Hello" - 4`
			`TypeError: unsupported operand type(s) for -: 'str' and 'int'`
			`Operator - cannot be applied to a string and an integer`
			
			`>>> a > 5`
			`NameError: name 'a' is not defined`

Error Types[2]

Name	Definition	Timing	Python Examples

Runtime Error	An error that occurs during the execution (runtime) of a program.	Happens when the code is run, and may halt the execution of the program	`>>> a = 5`
			`>>> b = 4 /(a - 5)`
			`ZeroDivisionError: division by zero`

Logical Error	An error that results from the code not conforming to its intended semantics due to, e.g., a formula being implemented incorrectly, a block of code being indented erroneously, the programmer misunderstanding and implementing incorrectly a requirement.	Happens when the code is run. May remain undiscovered.	`>>> duration_in_seconds = duration_in_hours * 100`
			
			`The formula for converting hours into seconds is incorrect`

Practice Problems

slido

What does the following code output?

ⓘ Start presenting to display the poll results on this slide.

Review Practice Problem 1

- **What does the following code output?**

```
x = 17 // 3

y = 9 % 5

print(x + y)
```

7.47
9
5 4
6
None of the above

slido

What does the following code output?

ⓘ Start presenting to display the poll results on this slide.

Review Practice Problem 2

- **What does the following code output?**

```
x = 7

x += 1


y = 2

y /= 4


print(x, y)
```

1 4
7 2
1 0
8 0.5
None of the above

slido

What is the outcome of running this code?

ⓘ Start presenting to display the poll results on this slide.

Review Practice Problem 3

- **What is the outcome of running this code?**

```
print("This is midterm 1')
```

'This is midterm 1'

This is midterm 1

Logical error

Semantic error

Syntax error

slido

What is the outcome of running this code?

ⓘ Start presenting to display the poll results on this slide.

Practice Problem 4

What is the outcome of running this code?

```
a = '100'
b = 50

subtractedNums = a - b

print(subtractedNums)
```

A. A logical error occurs
B. 50
C. A runtime error occurs
D. **A semantic error occurs**
E. A syntax error occurs

slido

What is the outcome of running this code?

ⓘ Start presenting to display the poll results on this slide.

Practice Problem 5

What is the outcome of running this code?

```
a = 100
b = 50

subtractedNums = a - b

print(subtractednums)
```

A. A logical error occurs
B. 50
C. A runtime error occurs
D. **A semantic error occurs**
E. A syntax error occurs

NOTE! print() must have a valid parameter inside

Learning Objectives

Upon completing this tutorial, the students will be able to:

1. use Wing 101 effectively
2. understand the concept of value / type / variable / operator / expression / error
3. recognize values / types / variables / operators / expressions / errors
4. define values (literals) / variables
5. use values, variables and operators (to define expressions)
6. evaluate simple expressions
7. fix basic errors

slido

Any questions?

ⓘ Start presenting to display the poll results on this slide.