



Introduction to Python

for applications to biomedical industries

BME 6303 | CRN 19454 | 3 credits

Asynchronous Learning | Lectures & Assignments Available Online

Office Hours, Mondays, 2:30 pm Central

[QutubLab.org/python](https://qutublab.org/python)



Instructor: Dr. Amina Ann Qutub

Amina.Qutub@utsa.edu

Additional Assistants:

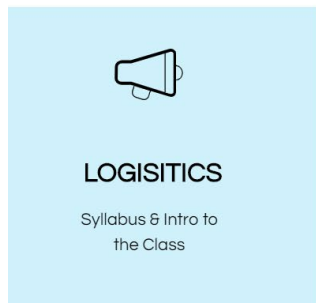
Byron Long (Byron.Long@utsa.edu)

Erin Pollet (Erin.Pollet@utsa.edu)

Jenny Brethen (Jennifer.Brethen@utsa.edu)

Office Hours, Mondays, 2:30 pm Central or by appointment

Accessing course materials



NOTE: Links to the course site are also available through UTSA's Blackboard. Three coding challenges and one final report are submitted through UTSA's Blackboard.

Main Course Site:

[QutubLab.org/python](https://qutublab.org/python)

Materials (Modules, Videos, Reading):

[QutubLab.org/pythonmaterials](https://qutublab.org/pythonmaterials)



Glossaries & Cheat Sheets

[QutubLab.org/pythonglossary](https://qutublab.org/pythonglossary)

Topics included: overall Python syntax, file input/output, data handling

What are included in modules?

16 Weekly Modules:

[QutubLab.org/pythonmaterials](https://qutublab.org/pythonmaterials)

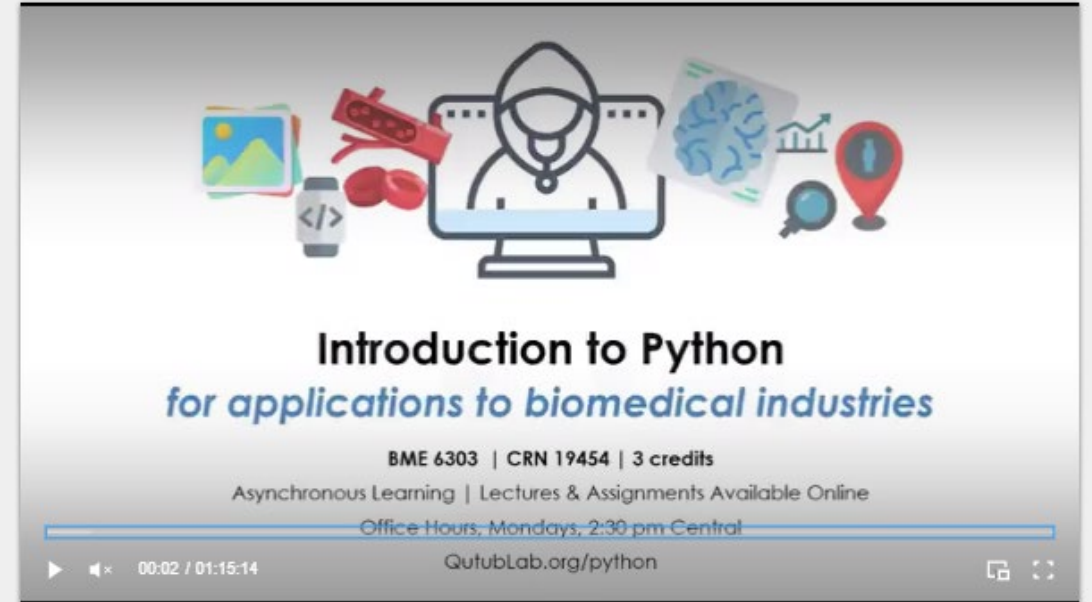
READ: Suggested reading

INTERACTIVE | WATCH: Videos, tutorial links, and/or slides

DO: Downloads, programming, coding challenges



Module 1 Welcome & Introduction



Weeks 1-2

Read

[Beginner's Guide to Python](#)

Watch

Introductory Video for Module 1 (above)

Do

1. Download & install [Python](#)
2. Download & install [PyCharm](#) (or other editor)
3. Bookmark [W3Schools Python Intro](#)
4. Sign up for [zyBooks](#) and subscribe to the Python book (Code: **UTSABME6303Fall2020**)
5. Try the first code "3 ways" as presented in the video lecture (@ ~28:20 into the video)

Recap for Module 1: Welcome & Introduction

Read

[Beginner's Guide to Python](#)

Interactive | Watch

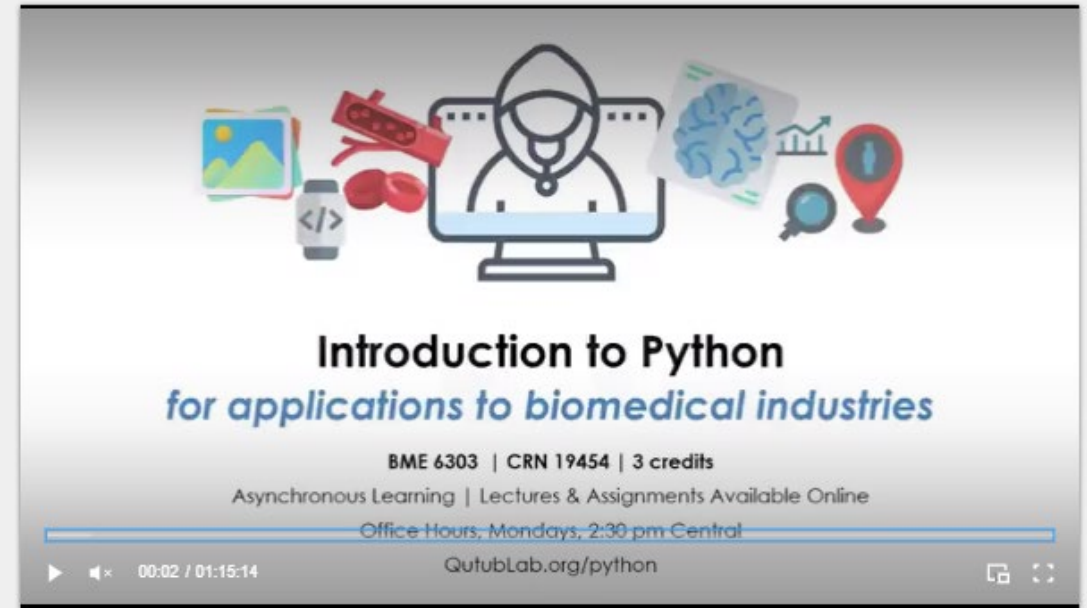
Introductory Video for Module 1

Do

1. Download & install [Python](#)
2. Download & install [PyCharm](#) (or other editor)
3. Bookmark [W3Schools Python Intro](#)
4. Optional: Sign up for [zyBooks and subscribe to the](#) Python book (Code: UTSABME6303Fall2020)
5. Try the first code "3 ways" as presented in the video lecture (@ ~28:20 into the video)



Module 1 Welcome & Introduction



Module 2:

Python Syntax (Data Types, Variables)

Read

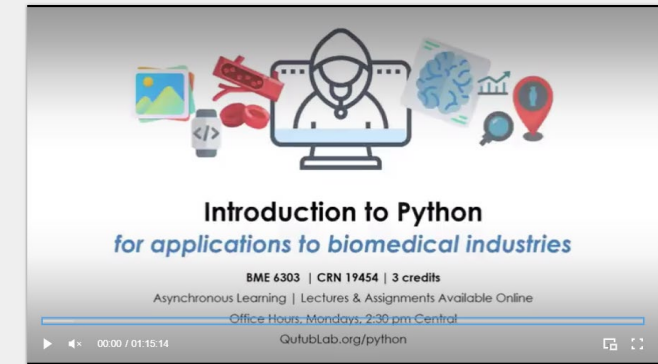
[W3Schools: Python Syntax Intro](#)

Interactive | Watch

Introductory Video for Module 2

Do

1. Complete Class Survey
2. Make & Upload on Dropbox an 15-30 sec Introduction Video about yourself
3. Complete [W3Schools Python Exercises 1-35, Syntax through Booleans](#)
4. Optional: Complete [zyBooks](#) Chapters 1 (Intro) & 2 (Variables & Expressions)
5. Try the “Joining a tiger team” syntax practice as presented in the video lecture



Weeks 2-3

Slides for
Module 2

Read
[W3Schools: Python Syntax Intro](#)

Module 3:

Python Operators

Read

[W3Schools: Python Operators Intro](#)

Interactive | Watch

Introductory Slides for Module 3

Do

1. Complete [W3Schools Python Exercises 35-39](#), Syntax through Booleans
2. Optional: Complete zyBooks Chapters 3 & 4
3. Try the “Smooth operator” operator practice as presented in the video lecture
4. Complete [Coding Challenge I](#). Submit **via Blackboard** for grading by next Tuesday, Sept 15th at 11:59 pm.

What are the goals of the modules?

Specific Objective I: To gain knowledge of the basic concepts of computer programming by learning the structure, syntax and implementation of the Python language.

Module 3: *Python Operators*

[QutubLab.org/pythonmaterials](https://qutublab.org/pythonmaterials)



Module 3:

Python Operators

[QutubLab.org/pythonmaterials](https://qutublab.org/pythonmaterials)



Module 3:

Python Operators

**We'll work in the Command Shell today –
please make sure to open the command shell**

QutubLab.org/pythonmaterials

Python Operators: Reading References

W3Schools Intro to Python Operators

https://www.w3schools.com/python/python_operators.asp

Automate the Boring Stuff (Author: Al Sweigart)

<https://automatetheboringstuff.com/2e/chapter1/>



Python Operators:

1. Arithmetic
2. Assignment
3. Comparison
4. Logical
5. Identity
6. Membership
7. Bitwise

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	x / y
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor division	$x // y$

Arithmetic Operators

Python Operators:

Assignment

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
=	x = 3	x = x 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

Assignment Operators

Python Operators:

1. Assignment
2. Comparison
3. Logical
4. Identity
5. Membership
6. Bitwise

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
=	x = 3	x = x 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

Assignment Operators

Python Operators:

1. Arithmetic
2. Assignment
3. Comparison
4. Logical
5. Identity
6. Membership
7. Bitwise

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Comparison Operators

Python Operators:

1. Arithmetic
2. Assignment
3. Comparison
4. Logical
5. Identity
6. Membership
7. Bitwise

Operator	Description	Example
and	Returns True if both statements are true	<code>x < 5 and x < 10</code>
or	Returns True if one of the statements is true	<code>x < 5 or x < 4</code>
not	Reverse the result, returns False if the result is true	<code>not(x < 5 and x < 10)</code>

Logical Operators

Operator	Description	Example
is	Returns True if both variables are the same object	<code>x is y</code>
is not	Returns True if both variables are not the same object	<code>x is not y</code>

Identify Operators

Python Operators:

1. Arithmetic
2. Assignment
3. Comparison
4. Logical
5. Identity
6. Membership
7. Bitwise

Operator	Description	Example
in	Returns True if a sequence with the specified value is present in the object	x in y
not in	Returns True if a sequence with the specified value is not present in the object	x not in y

Membership Operators

Operator	Name	Description
&	AND	Sets each bit to 1 if both bits are 1
	OR	Sets each bit to 1 if one of two bits is 1
^	XOR	Sets each bit to 1 if only one of two bits is 1
~	NOT	Inverts all the bits
<<	Zero fill left shift	Shift left by pushing zeros in from the right and let the leftmost bits fall off
>>	Signed right shift	Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off

Bitwise Operators

Hexadecimal Numbers

Decimal Number	4-bit Binary Number	Hexadecimal Number
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C

Reference:

https://www.electronicstutorials.ws/binary/bin_3.html

Python Operators:

Arithmetic

```
Python 3.8 (32-bit)
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 2-1
1
>>> 40 + 2
42
>>> 4/5
0.8
>>> x = 4/5
>>> print(x)
0.8
>>> y = 0.8
>>> x%y
0.0
>>> y = 0.3
>>> x % y
0.20000000000000007
>>> x ** y
0.9352484478226214
>>> x/y
2.666666666666667
>>> x//y
2.0
>>> z = x + y**x
>>> z
1.1816778909618177
>>>
```