Welcome to our webinar: Getting Started with Python in the Classroom

- Who is in the room?
- Please introduce yourself in the chat window:
  - Your name
  - Your position
  - Your city/state

We want to HEAR from you! Please choose the first option in the Audio Conference window (Dial-out) in order to communicate verbally. If unable to receive an incoming call, please choose one of the other options listed.
Getting Started with Python in the Classroom

Jennifer Vermillion
October 23, 2019
Webinar outline

• **Who?** Who’s here tonight?
• **Why?** Why Python?
• **What?** Basic building blocks in Python
• **Where?** A “cozy place” to program
• **How?** Ideas for teaching Python
• **What’s Next?** Resources to keep you going
POLL

- What is your experience with Python?
Programming languages

Why Do You Want to Learn to Code?

- Be a professional developer
- Run a startup
- Build something fun
- Have another skill
Why Python?

• Great first language
• Powerful, yet easy to learn
• Relevance

Image Source: https://www.merixstudio.com/blog/is-python-best-programming-language-your-startup/
Python 3

Guido van Rossum
1956-Present

Van Rossum is best known as the author of the Python programming language.

He has also worked for Google and Dropbox.
Getting started: print statements and strings

Code —> 1 print("Hello world!")

Output —> Hello world!

Getting started: print statements and strings

Code —>

```python
1    print("After the keyword print")
2    print("anything you enter")
3    print("inside the parentheses")
4    print("will be displayed!")
```

Output —>

```
After the keyword print
anything you enter
inside the parentheses
will be displayed!
```
## Types of data

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>Characters inside quotation marks</td>
<td>“This is a 4 string”</td>
</tr>
<tr>
<td>Integer</td>
<td>Whole Numbers</td>
<td>153980</td>
</tr>
<tr>
<td>Float</td>
<td>Numbers with decimals</td>
<td>1.345</td>
</tr>
<tr>
<td>Boolean</td>
<td>True or False Value</td>
<td>5 &lt; 4</td>
</tr>
</tbody>
</table>
Variables

• A **variable** name must start with a letter or the underscore character.

• A **variable** name can only contain alphanumeric characters and underscores (A-z, 0-9, and _ )

• **Variable** names are case-sensitive (age, Age and AGE are three different **variables**)
Assignment statement

• An assignment statement is used to assign a value to a variable.

• Variables can contain
  – strings (ex: “this is a string”)
  – integers (ex: 4)
  – floating point numbers (ex: 4.34346)
  – and more
Assignment statement

Code —>  
1 x = 5  
2 y = 4  
3 print (x,y)

Output —>  
5 4
Input

Code —> 1  User_name = input("What is your name? ")

Output —> What is your name?
Input

Code —►

```
1   User_name = input("What is your name? ")
2   print ("Hello", User_name)
```

Output —►

```
What is your name? 
What is your name? Jennifer 
Hello Jennifer
```
Comments (#)

#This is a comment
#Python skips over comments
Basic math: Add

**Code**

```python
1  #Add
2  x = 5
3  y = 2
4
5  My_Sum = x+y
6  print(My_Sum)
7  #or
8  print(x+y)
```

**Output**

7

7
Basic math: Subtract

**Code**

1. `#Subtract`
2. `x = 5`
3. `y = 2`
4. `My_Difference = x - y`
5. `print(My_Difference)`
6. `#or`
7. `print(x - y)`

**Output**

3

3
Basic math: Multiply

**Code**

1. `#Multiply`
2. `x = 5`
3. `y = 2`
4. `My_Product = x*y`
5. `print(My_Product)`
6. `#or`
7. `print(x*y)`

**Output**

```
10 10
```
Basic math: Divide

Code

```python
# Divide
x = 5
y = 2

# float division
print(x/y)

# integer division
print(x//y)
```

Output

```
2.5
2
```
Conditional expressions

Code

1 #Comparison Operators
2 #Greater Than >
3 #Greater Than or Equal To >=
4
5 if 5 > (2+3):
6  print ("Bam!")

Output
Conditional expressions

**Code**

```python
1 #Comparison Operators
2 #Less Than  <
3 #Less Than or Equal To  <=
4
5 if 5 <= (2+2):
6     print ("Bam!")
```

**Output**

> I
Conditional expressions

**Code**

```python
#Comparison Operators
# are these things not equal? !=

x = 4
y = 3
if x != y:
    print ("They are not equal!")
```

**Output**

They are not equal!
Conditional expressions

**Code**

```python
# Comparison Operators
# are these things equal? ==
x = 4
y = 4
if x == y:
    print("They are equal!")
```

**Output**

They are equal!
Loops

**Code**

```python
# Counted Loop

for looper in range(3):
    print("I'm a loop!")
```

**Output**

```
I'm a loop!
I'm a loop!
I'm a loop!
```
Loops

**Code**

```
1    #Conditional Loop
2
3    looper = 5
4    while looper > 0:
5        print(looper)
6        looper = looper - 1
```

**Output**

```
5
4
3
2
1
```
Python Turtle
Any questions so far?
Getting started: Where can I program?

Online integrated development environment

From Wikipedia, the free encyclopedia

An **online integrated development environment**[^1] (also known as a Web IDE[^2] or Cloud IDE[^3]), is a browser based integrated development environment. A online IDE can be accessed from a web browser, such as Google Chrome or Internet Explorer, enabling software development on an low-powered devices that are normally unsuitable. An online IDE does not usually contain all of the same features as a traditional, or desktop, IDE, although all of the basic IDE features, such as a [source code editor with syntax highlighting](http://example.com). Integrated [version control](http://example.com) and [Read-Eval-Print Loop(REPL)](http://example.com) may also be included.
Getting started: Where can I program?

repl.it
Getting started: Where can I program?

trinket
Teaching Ideas
Warm-ups: Predict the output

```python
x = 7
y = 3
z = x % y
if z > y:
    print("Twizzlers")
elif z < x:
    print("Fiddlesticks")
```
Find the error(s)

6.) Describe the error in the following code (assume indentation is correct):

```python
if a = b:
    print("done!")
else a > b:
    print("try again")
```

Error: ___________________________________________________________
# Highlight your code

## Color Coding: Python Battleship (Daily Warm-Up)

Highlight each of the programming structures listed below that you find in the Python Battleship code. Note that not all of the code will be highlighted.

<table>
<thead>
<tr>
<th>Lists</th>
<th>Soft Pink</th>
</tr>
</thead>
<tbody>
<tr>
<td>List Methods</td>
<td>Soft Teal</td>
</tr>
<tr>
<td>Function Definition</td>
<td>Soft Blue</td>
</tr>
<tr>
<td>Function Calls</td>
<td>Bright/Neon Orange</td>
</tr>
<tr>
<td>Loops</td>
<td>Yellow</td>
</tr>
<tr>
<td>Conditional Tests</td>
<td>Soft Purple</td>
</tr>
</tbody>
</table>

```python
# game Board
board = ['•' for _ in range(10)]

# Player Variables
players = ['player_1', 'player_2']

# player turn
for turn in range(4):
    for player in players:
        # requesting row and column from player
        guess_row = int(input(f"Guess Row for {player}: ")) - 1
        guess_col = int(input(f"Guess Col for {player}: ")) - 1

        # if guessed the correct cell
        if guess_row == cell_row and guess_col == cell_col:
            print("Congratulations! You sank my battleship!")
            break
        else:
            board[guess_row] = 'x'
            print(f"You missed {player}!")

# print the board
for row in board:
    print(row)
```

Imagine Code Create (H)
Team Project: Create Your Own Personality Quiz

Directions. Your task is to design a clever interactive personality quiz as a team. This may seem like a daunting task, but we can break down the steps using decomposition. Decomposition is one of the cornerstones of computer science. It involves breaking down a complex problem or system into smaller parts that are more manageable and easier to solve. The smaller pieces can then be solved or designed individually.

<table>
<thead>
<tr>
<th>Brainstorm possible themes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine the spectrum of possible</td>
</tr>
</tbody>
</table>
Programming sprints

- Stand up in a circle and keep it short!
- Q: What did you accomplish?
- What challenges are you experiencing?
Resources
Resources

![Book Covers]

- **Hello World!**
  - Computer Programming for Kids and Other Beginners
  - Warren and Carter Sande
  - Manning

- **Python Programming:**
  - An Introduction to Computer Science
  - Second Edition
  - John Zelle
  - Franklyn Beadle & Associates Inc.
  - Independent Publishers Since 1985
Upcoming activities

• *What Does a CS Classroom Look Like in Elementary School* with Lisa Rode, Tuesday, November 12 at 7:30pm ET/4:30pm PT.

• Stay tuned for upcoming CSEdWeek activities, December 9 through 13 on the community site and Twitter account!
Are you a member of CS for All Teachers?

Membership is FREE! Sign up in order to:

- Connect with other CS teachers
- Participate in free webinars
- Access free resources
- Learn more about ECS and CSP curricula
- Download the latest ECS assessments

https://csforallteachers.org/user/register
CS for All Teachers

Email: csforallteachers@air.org
Website: https://csforallteachers.org/
Twitter: @CSforAllTchrs