

Python Workshop Day 1

Coding Pals | UBC Edith
Lando

Who are we?



Gary Li

Grade 12 at St. George's School

Favourite Food: Ramen

Hobby: Jazz drumming

Favourite Brand: Razer

Davis Clark

Grade 12 St. George's School

Favourite Food: Sushi

Hobby: Piano

Favourite Brand: Google

What is Computer Programming?

Computer programming is a way to give computers instructions about what they should do.

Types of Languages

- Python
- Java
- HTML
- Javascript
- CSS
- C++

Why learn Python?

1. Easy to learn and understand. It is user-friendly, simple, and innovative
2. Gives learners a boost when learning more programming languages in the future.
3. Big companies use Python: Spotify, Netflix, Google Instagram etc.

“Hello, World”

- C

```
#include <stdio.h>

int main(int argc, char ** argv)
{
    printf("Hello, World!\n");
}
```

- Java

```
public class Hello
{
    public static void main(String argv[])
    {
        System.out.println("Hello, World!");
    }
}
```

- now in Python

```
print "Hello, World!"
```

What is Pseudocode?

- Informal description of steps
- Easily translatable to English
- Used to plan steps

The programming thought process

1. Open can using can opener
2. Pour contents of can into saucepan
3. Place saucepan on stove burner
4. Turn on correct burner
5. Stir stew until warm

How to approach a problem

1. Analyze the problem
2. Design a solution
3. Program the solution
4. Test the program

Analyze the problem

- Make sure you understand the problem
- Make sure you have all of the necessary information to solve the problem
 - (ie. variables, functions, etc.)
- Think of the problem like a computer
 - (ie. Why is 7 prime?)

Design a Solution

- Writing the solution is the most important part in solving a problem
- Find out what steps are necessary to solve the question

Example:

Find the sum of two numbers

1. Get numbers a,b
2. $\text{Sum} = a+b$
3. Print Sum

Example: Find the Perimeter of a triangle

1. Get sides a , b , c
2. Perimeter = $a+b+c$
3. Print out Perimeter

Example: Delete the number 3 from a list

1. Get List [1,4,3,7,3,6,3,9]
2. Look through each item in the list
 - a. If number is 3
 - i. Delete
 - b. If number is not 3
 - i. Ignore
3. Get new list [1,4,7,6,9]

What is Syntax?

The syntax of the Python programming language is the set of rules that defines how a Python program will be written and interpreted.

Why learn Syntax?

1. Base rules for Python
2. Everything builds up from Syntax
3. Important foundation for any programming language

Comments

Python has commenting capability for the purpose of documenting code.

3 main purposes:

1. Comments can be used to explain Python code.
2. Comments can be used to make the code more readable.
3. Comments can be used to prevent execution when testing code

Comments start with a #, and Python will ignore it while executing.

`#this is a comment`

`print("Hello World")`

`#print("Hello World")`

`#print("Hello World")`

`#print("Hello World")`

What is a Variable?

A variable is a storage for some value

A variable is created the moment you first assign a value to it.

Variables can have many different values; we call these different values “datatypes”

Data Types Pt.1

- In programming, data type is an important concept.
- Variables can store data of different types, and different types can do different things.

Python has many data types built-in by default, but we will only look at the most basic ones:

Text Type:

`str`

Numeric Types:

`int, float`

Sequence Types:

`list`

Boolean Type:

`bool`

Data Types Pt.2

["apple", "banana", "cherry"] #list

“Hello World” #str, strings are arrays of bytes representing Unicode characters.

5 #int, integer numbers

5.55 #float, decimal numbers ex. 5.00

[1,2,3,4,5] #list

True #bool, true or false

How do we create a variable?

By setting a variable name on the left with an equal sign in the middle and the value on the right, we can create a variable

Examples:

`X = 10 #int`

`CP = "computer programming" #string`

`Cats = True #boolean`

`Y = 11.11 #float`

Why use variables?

If use the same values in multiple places but want to change it later, it can be a hassle changing each value. We can simply use a variable instead and just change the value of the variable once to implement these changes throughout our entire code.

Being able to reference variables in multiple places is what makes them so useful.

For example, we can directly print variables to the console:

```
name = "Davis"
```

```
print(name) #prints "Davis" on the console
```

```
name = "Gary" #changing the value of the variable called name
```

```
print(name) #prints "Gary" on the console
```

How do we add numerical variables together?

Mathematical rules still apply in Python!

If our variables are both numbers of the same data type, we can add them using a “+” sign:

```
X = 10
```

```
Y = 5
```

```
print(X+Y) #prints 15
```

(We can also add values to a single variable too)

Python Arithmetic Operators

Arithmetic operators are used with numeric values to perform common mathematical operations:

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

% = remainder

// = removes decimals (not rounding!)

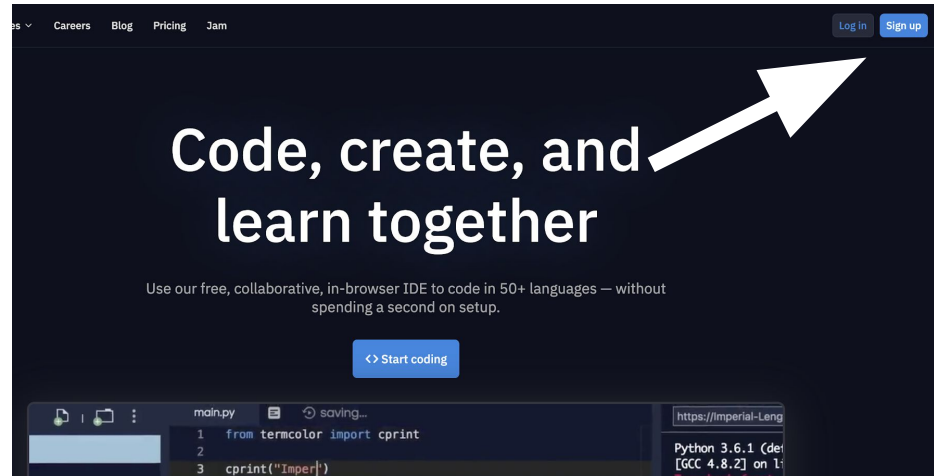
Python Assignment Operators

Assignment operators are used to assign values to variables:

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

How to use repl.it Part 1

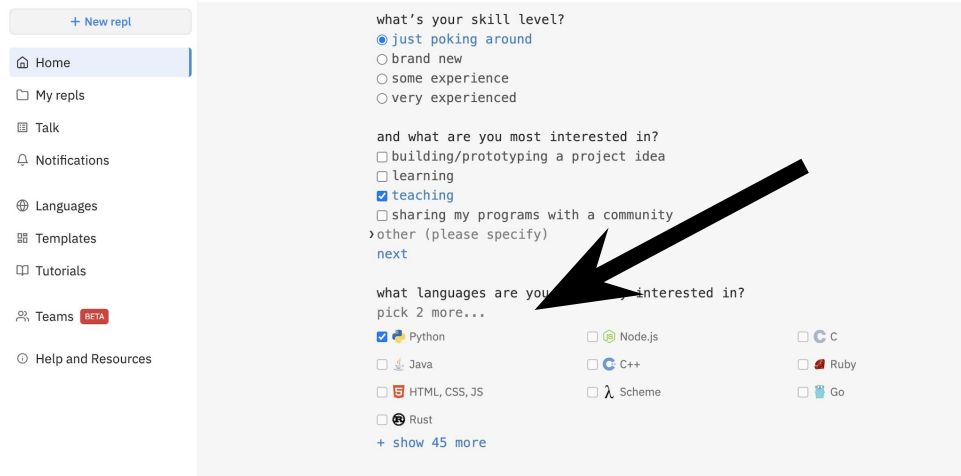
1. Go to <https://repl.it/>
2. Select “Sign up” on the top right corner (white arrow)
3. Sign up with Google, Github, Facebook, or make your own account



How to use repl.it Part 2

4. Confirm/fill in your profile information

(make sure to select Python as one of your primary languages)



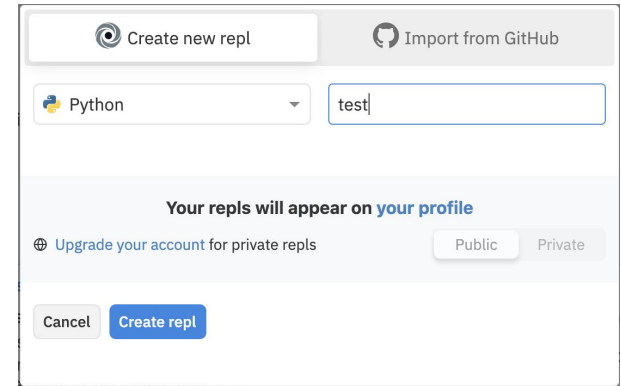
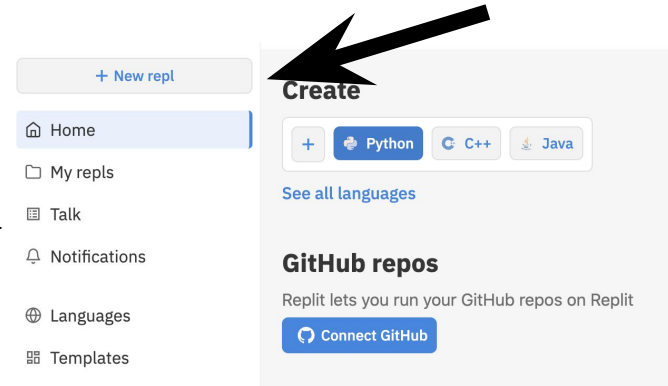
The screenshot shows the repl.it profile setup interface. On the left is a navigation menu with options: Home (selected), My repls, Talk, Notifications, Languages, Templates, Tutorials, Teams (with a 'beta' badge), and Help and Resources. The main content area contains the following sections:

- what's your skill level?**
 - just poking around
 - brand new
 - some experience
 - very experienced
- and what are you most interested in?**
 - building/prototyping a project idea
 - learning
 - teaching
 - sharing my programs with a community
 - other (please specify)
 - next
- what languages are you interested in?**
pick 2 more...
 - Python
 - Java
 - HTML, CSS, JS
 - Rust
 - Node.js
 - C++
 - Scheme
 - C
 - Ruby
 - Go[+ show 45 more](#)

How to use repl.it Part 3

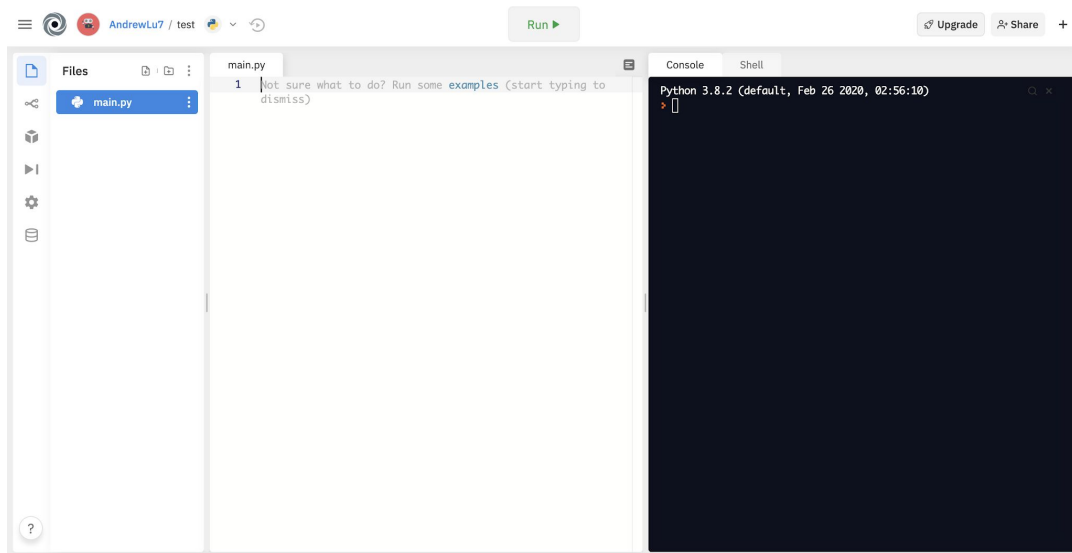
5. Go to the top left hand corner and click new repl.

6. Select Python from drop down menu for your repl & press create repl.



How to use repl.it Part 4

You can start programming now! The code goes in the “main.py” file and the output will be shown on the “console”. Play around with variables and print statements with what we’ve learned so far!



What are If statements?

In Python, If Statement is used for decision making. It will run the body of code only when IF statement is true.

When you want to justify one condition while the other condition is not true, then you use "if statement".

Python Indentation

Indentation refers to the spaces at the beginning of a code line.

Python uses indentation to indicate a block of code. Everything that is indented is considered part of the if statement

Your IDE will do this automatically.

You must have a colon at the end of the if statement

```
if 5 > 2:  
    print("Five is greater than two!")
```

if Statements

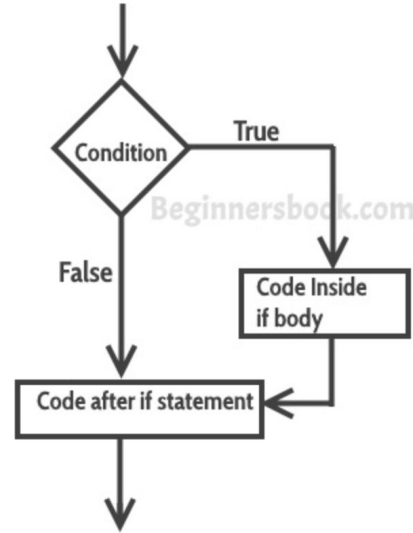
Example:

```
flag = True
```

```
if True:  
    print("Welcome")  
    print("To")  
    print("Coding Pals")
```

Output:
Welcome
To
Coding Pals

If statement flow diagram



elif Statements

The `elif` keyword is python's way of saying "if the previous conditions were not true, then try this condition".

```
a = 33
```

```
b = 33
```

```
if b > a:
```

```
    print("b is greater than a")
```

```
elif a == b:
```

```
    print("a and b are equal")
```

else Statements

The `else` keyword catches anything which isn't caught by the preceding conditions.

```
a = 200
```

```
b = 33
```

```
if b > a:
```

```
    print("b is greater than a")
```

```
elif a == b:
```

```
    print("a and b are equal")
```

```
else:
```

```
    print("a is greater than b")
```

Nested if Statements

You can have `if` statements inside `if` statements, this is called *nested if* statements.

```
x = 41

if x > 10:
    print("Above ten,")
    if x > 20:
        print("and also above 20!")
    else:
        print("but not above 20.")
```

Python Comparison Operators

Comparison operators are used to compare two values:

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

Python Logical Operators

Logical operators are used to combine conditional statements:


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>

DMOJ

- Easy access to problems
- Online grader for many languages
- [Dmoj.ca](https://dmoj.ca)

How to submit for grading:

- Copy and paste your code into the blue submit button in the problem page
- Select Python 3 in the tab below
- Press submit



The screenshot shows a portion of a DMOJ problem page. On the right side, there is a blue button labeled "Submit solution". Below it, there are links for "My submissions", "All submissions", and "Best submissions". Further down, the problem's statistics are listed: "Points: 3", "Time limit: 1.0s", and "Memory limit: 512M". On the left side, there is a text area containing the problem description, which includes a formula $\cdot 3 \times L$ and a question about Barley's happiness. At the bottom of the page, there is a language selection tab for "Python 3 (python3 3.7.3 - 3.8.3)".

Example question: CCC'2019 Junior 1

- We will do a step to step tutorial right now
- [Link](#) to problem

What are For and While Loops?

In the language of Python, “While loops” are loops that execute a certain command *while* a condition is true. This will be helpful when finding answers that match several different conditions at once.

“For loops”, on the other hand, are loops used to go through a sequence. You will later learn what “strings” and “lists” are, and “For” loops are useful to iterate through those strings and lists.

Why use loops? They help us repeat processes without copying the same code over and over

Python Indentation

Indentation refers to the spaces at the beginning of a code line.

Python uses indentation to indicate a block of code. Loops also need colons afterwards, and everything intended is considered inside the loop

Your IDE will do this automatically.

```
if 5 > 2:  
    print("Five is greater than two!")
```

How to use While Loops

In order to use While loops, you need to declare it in a way similar to If statements. The most basic method requires an integer, which is increased each time the loop repeats until it no longer matches the loop condition.

In the following code, the number 4 is printed out 4 times.

```
n = 1

while n < 5:

    print(4)

    n += 1
```

How to use For Loops

The most basic way to use “For loops” is to use it with the `range()` function. `range()` starts at 0 by default, increases by 1, and returns each number until an end number. Remember: Python numbering makes it so that the real end number is one less!

The following code prints all the numbers from 0 to 5.

```
for x in range(6):
```

```
    print(x)
```

```
for x in range(0,6):
```

```
    print(x)
```

How to use For Loops

With the `for` loop we can execute a set of statements, once for each item in a list, tuple, set etc.

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)
```

```
for x in "banana":  
    print(x)
```

Break Statements

Break statements can help stop loops, even though the initial condition may still be true. This can be used in both “For” and “While” loops.

Below is a break statement within a “For” loop.

```
for x in range(6):  
    print(x)  
    if x == 3:  
        break
```

```
i = 1  
while i < 6:  
    print(i)  
    if i == 3:  
        break  
    i += 1
```

Continue Statements

Continue statements are statements that allow code to “skip” the following code and continue to the next number/thing in the list.

In the following code, the number 3 is skipped in a “While” loop.

```
n = 1
while n < 5:
    n += 1
    if n == 3:
        continue
    print(n)
```

```
fruits = ["apple", "banana",
"cherry"]
for x in fruits:
    if x == "banana":
        continue
    print(x)
```

Guide to Solving a Junior 1 Problem

- **Determine your input:**
 - Is it a string? In this case: `x = input()` or `x = str(input())`
 - Is it a integer? In this case: `x = int(input())`
- **Determine what combination if statements are needed:**
 - `if/elif/else`
 - `for/while` loops
- **Use a variable or counter to make the calculation/operation**
- **Use `print()` to finish the problem**

CCC'2014 Junior 2

- We will do a step to step tutorial right now
- [Link](#) to problem

Homework:



1. Make an account at dmoj.ca
 - a. Go to the problems tab on the top left side
 - b. Go to problem search on the right side
 - i. Search “ccc13j1”
 - ii. Search “ccc16j1”
 - iii. Search “ccc20j2” (slightly harder)
2. Do all the problems listed above and submit them for grading. Don't worry if you don't get them all right, we will review them next class.

The image shows a screenshot of the "Problem search" interface on the DMOJ website. It features a search input field containing the text "ccc18j1". Below the input field are three checkboxes: "Full text search", "Hide solved problems", and "Show problem types", all of which are currently unchecked. Underneath these is a "Category" dropdown menu set to "All". Below the dropdown is a "Point range" slider, which is currently set between 1 and 50. At the bottom right of the search panel are two buttons: "Go" and "Random".