# HackerFrogs Afterschool
# Python Programming Basics: Part 2

Class:
Programming (Python)

Workshop Number:
AS-PRO-PY-02

Document Version:
1.5

Special Requirements:
None

# What We Learned Last Time

This is the second workshop for intro Python programming.

In the previous session we learned about the following Python concepts...

# Hello, World!


We wrote the **Hello, World!** program in Python, which is the first program students write for any computer language.

# The Print Function

```
>>> print("This is how you print text in Python!")
This is how you print text in Python!
```

We learned how to use the Python print function, which prints information to the console and is common to almost all programs written in Python.

# Variables

```
>>> name = "HackerFrogs"
>>> greetings = "Hello, "
>>> print(greetings + name)
Hello, HackerFrogs
```

We learned about programming variables, named storage locations which hold specific values, which can be referenced multiple times in a program.

# Data Types

```
>>> pi = 3.14
>>> number = 1337
>>> name = "shyhat"
>>> print(type(pi),type(number),type(name))
(<class 'float'>, <class 'int'>, <class 'str'>)
```

We learned about Data Types, including **strings** (non-numeric text), **integers** (whole numbers), and **floats** (numbers which include decimal places).

# This Workshop's Topics

Part 1: Lists

Part 2: Arithmetic Operations

# Part 1: Lists

Lists are part of what Python categorizes as collections, variables that hold one or more values, which can be strings, numbers, or a mixture of different data types.

# Lists

An example list
in Python could look
like the following:

# Lists

```
vegetables = ["carrots", "lettuce", "beets"]
```

or

```
related_to_pie = ["round", 3.14, "delicious"]
```

As we can see by the example, a list is defined as a number of objects (separated by commas ' , ') between a pair of square brackets ' [ ] '.

# List Indexes

```
fruits = ['apple', 'orange', 'grapes']
fruits[0] = 'apple'
fruits[1] = 'orange'
fruits[2] = 'grapes'
```

Items in lists occupy ordered positions within it, called indexes, with the first item in the index 0 position, the second item in the index 1 position, etc.

# List Indexing

```
fruits = ['apple', 'orange', 'grapes']
fruits[0] = 'apple'
fruits[1] = 'orange'
fruits[2] = 'grapes'
```

List indexing allows for the selection of specific items in a list by referring to them by the position in which they appear in the list.

# List Indexing

```
fruits = ['apple', 'orange', 'grapes']
fruits[0] = 'apple'
fruits[1] = 'orange'
fruits[2] = 'grapes'
```

To use list indexing, use the name of the list, directly followed by a pair of square brackets ' [ ] ', with the index (position) of the desired item inside the brackets.

# List Indexing

```
trees = ['pine', 'elm' , 'cedar']

print(trees[0])
```

Look at the Python code above. Inside the print function, the trees list's index 0 is referenced. Which list item will be print output?

# List Indexing

```
trees = ['pine', 'elm' , 'cedar']

print(trees[0])
```

The output should be **pine**, since pine is the first item on the list. Note that counting positions in indexing start from zero, not one. In this example, `trees[2]` refers to **cedar**, since cedar is the number 2 item on the list if we count from zero.

# List Indexing – Changing Index Items

```
trees = ['pine', 'elm', 'cedar']
print(trees[0])
trees[0] = 'oak'
print(trees[0])
```

```
pine
oak
```

An important thing to remember when working with lists is that the value of items in lists can changed after initial assignment.

# Data types, Classes, and Objects

```
empty_list = []
empty_list_type = type(empty_list)
print(empty_list_type)
```

```
<class 'list'>
```

Every data type in Python, (lists, strings, integers, etc) belongs to a corresponding class, and each instance of a class is an object.

# What are Methods?

```
>>> vegetables = ["carrots", "lettuce", "beets"]
>>> vegetables.append("radishes")
>>> print(vegetables)
['carrots', 'lettuce', 'beets', 'radishes']
```

Methods are class-specific functions, e.g., there list methods, string methods, integer methods, etc. Methods perform actions on the object.

# List Methods

```
>>> vegetables = ["carrots", "lettuce", "beets"]
>>> vegetables.append("radishes")
>>> print(vegetables)
['carrots', 'lettuce', 'beets', 'radishes']
```

Methods are performed by indicating the name of the object (the list), then a dot ( . ), then the name of the method, followed by a pair of parentheses, which contain any arguments required for that method.

# List Methods - Append

```
>>> vegetables = ["carrots", "lettuce", "beets"]
>>> vegetables.append("radishes")
>>> print(vegetables)
['carrots', 'lettuce', 'beets', 'radishes']
```

The list method we will learn in this workshop is the **append** method. The append method will add an item to the end of the specified list.

Take the following code as an example:

# List Methods - Append

```
friends = ['Alex', 'Barb']

friends.append('Carl')

print(friends)
```

The output should be:

```
['Alex', 'Barb', 'Carl']
```

# Del – Deleting Items from Lists

```
friends = ['Alex', 'Barb', 'Carl']
del friends[0]
print(friends)
```

```
['Barb', 'Carl']
```

Deleting items from lists can be done in a few different ways, but if you know the list index of the item you want to delete, you can use the **del** keyword

# Del – Deleting Items from Lists

```
friends = ['Alex', 'Barb', 'Carl']
del friends[0]
print(friends)
```

```
['Barb', 'Carl']
```

To delete an item, use the **del** keyword, then the name of the list, a pair of square brackets, and the index of the item to be deleted inside the brackets

# Lists Exercise

Let's cover the next few exercises on the Learnpython.org website:

https://learnpython.org/en/Lists

# Part 1 – Lists Quiz - Q1

Which of the following statements about Python lists is true?

A) When printing a list, all items in the list must be printed

B) It's not possible to add items to a list after list creation

C) Counting items in a list starts from zero, not one

D) Items in a list cannot be changed after they've been added

# Part 1 – Lists Quiz - Q1

Which of the following statements about Python lists is true?

A) When printing a list, all items in the list must be printed

B) It's not possible to add items to a list after list creation

C) Counting items in a list starts from zero, not one

D) Items in a list cannot be changed after they've been added

# Part 1 – Lists Quiz - Q2

Which of the following code will add an string item, apple, to a list named fruits?

A) fruits + item.add(apple)

B) fruits.append('apple')

C) fruits + ['apple']

D) None of the above

# Part 1 – Lists Quiz - Q2

Which of the following code will add an string item, apple, to a list named fruits?

A) fruits + item.add(apple)

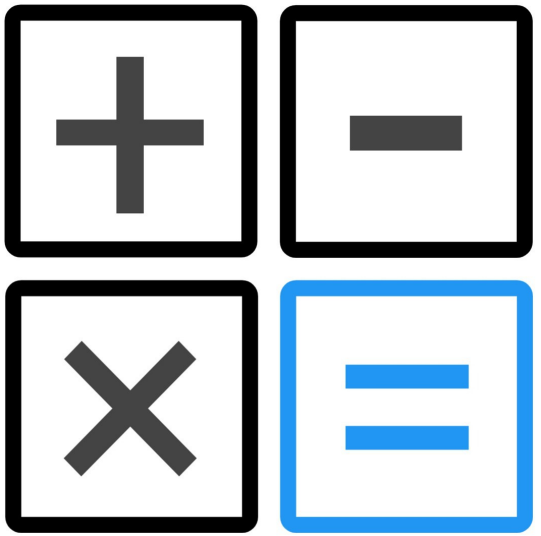
B) fruits.append('apple')

C) fruits + ['apple']

D) None of the above

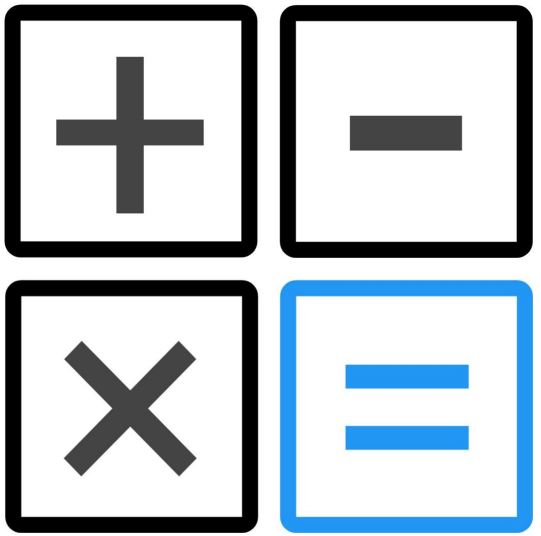
# Part 2: Basic Operators

In programming terms, operators are used to perform an operation on a piece of data.



# Basic Operators

In this lesson, we will learn about basic arithmetic operators that are used for mathematical addition, subtraction, division, multiplication, and a couple of others.



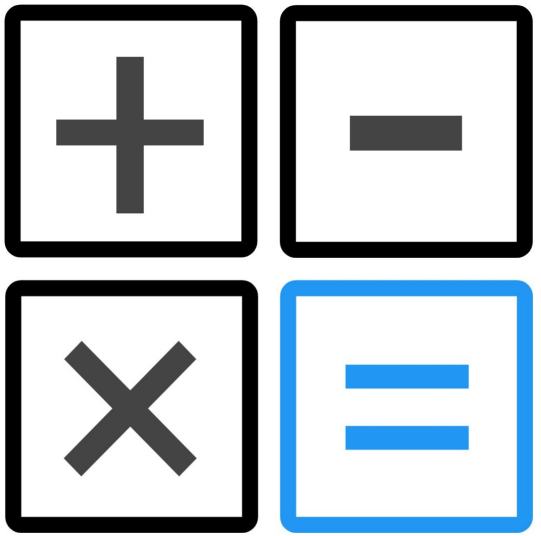
# Basic Operators

Python uses the following operators (symbols) for basic arithmetic:

| Operator Name | Symbol | Example |
|---|---|---|
| Addition | + | x + y |
| Subtraction | - | x - y |
| Multiplication | * | x * y |
| Division | / | x / y |
| Exponentiation | ** | x ** y |

# Basic Operators

The arithmetic operations that occur in Python follow the PEMDAS order of operations.



# Basic Operators

(P)arentheses
(E)xponents
(M)ultiplication
(D)ivision
(A)ddition
(S)ubtraction

# Basic Operators

(P)arentheses
(E)xponents
(M)ultiplication
(D)ivision
(A)ddition
(S)ubtraction

# Basic Operators

(P)arentheses
(E)xponents
(M)ultiplication
(D)ivision
(A)ddition
(S)ubtraction

# Basic Operators

(P)arentheses
(E)xponents
(M)ultiplication
(D)ivision
(A)ddition
(S)ubtraction

# Basic Operators

(P)arentheses
(E)xponents
(M)ultiplication
(D)ivision
(A)ddition
(S)ubtraction

# Basic Operators

(P)arentheses
(E)xponents
(M)ultiplication
(D)ivision
(A)ddition
(S)ubtraction

# Basic Operators

(P)arentheses
(E)xponents
(M)ultiplication
(D)ivision
(A)ddition
(S)ubtraction

# Basic Operators

In summary, all operations enclosed in (p)arentheses are performed first, then all (e)xponent operations are resolved, followed by (m)ultiplication, (d)ivision, (a)ddition, then (s)ubtraction operations at the end.

# Basic Operators

Given the following equation in Python syntax:

```
(4 / 2) * 2 ** 2 / 4 + 2 - 4
```

First the division operation inside of the parentheses (4 divided by 2) is performed. Second, the exponent operation (2 ** 2, which is Python's syntax for “two to the second power”, is resolved. Third, the multiplication (2 times 4) is resolved.

# Basic Operators

Given the following equation in Python syntax:

```
8 / 4 + 2 - 4
```

Forth, the division (8 divided by 4) is resolved. Fifth, the addition (2 + 2) is resolved. Finally, the subtraction (4 - 4) is resolved, which sets our final result to 0 (zero).

# Basic Operators - Modulo

A common operator used in programming is the **modulo** operator ( % )

```
5 % 2 = 1
8 % 5 = 3
```

A modulo operation returns the remainder after one number is divided by another. The resulting number after a modulo operation is called the **modulus**

# Basic Operators - Modulo

```
n % 2
```

A typical use for the modulo operation is to determine whether a number is even or odd by performing a modulo operation on it with the number 2. If the result is 1, the number is odd, and if the result is 0, the number is even

# Basic Operators Exercise

Let's cover the next few exercises on the Learnpython.org website:

https://learnpython.org/en/Basic_Operators

# Part 4 – Basic Operators Quiz - Q1

Which of the following methods will allow us to print a string and an integer with the same print function?

A) No special method need be taken to mix a string and an integer with the print function

B) The integer must be converted to a string to mix integers and strings in a print function

C) The string must be converted to an integer to mix integers and strings in a print function

D) You cannot print a string and integer with print

# Part 4 – Basic Operators Quiz - Q1

Which of the following methods will allow us to print a string and an integer with the same print function?

A) No special method need be taken to mix a string and an integer with the print function

B) The integer must be converted to a string to mix integers and strings in a print function

C) The string must be converted to an integer to mix integers and strings in a print function

D) You cannot print a string and integer with print

# Part 4 – Basic Operators Quiz - Q2

Which of the following math equations will let us determine if a number is even or odd?

A) n / 2

B) n ** 2

C) n % 2

D) n - 2

# Part 4 – Basic Operators Quiz - Q2

Which of the following math equations will let us determine if a number is even or odd?

A) n / 2

B) n ** 2

C) n % 2

D) n - 2

# Extra – the chr() Function

In Python, the chr() function is used to convert numbers to their Unicode equivalents. Unicode encoding is what lets computers turn the bytes into letter and numbers for the monitors and other screens.

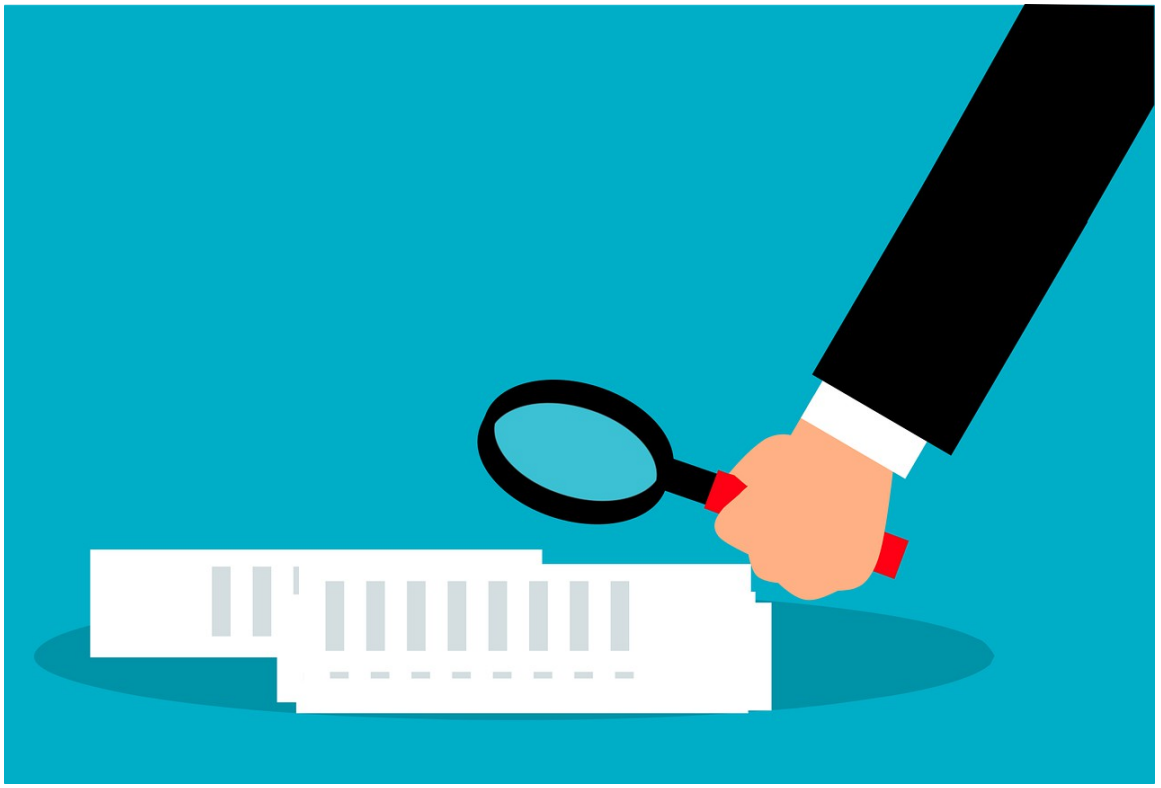
For example, if we run the `chr(65)` function, the result will be the uppercase letter A.

# Workshop Review Exercise

Before finishing today's workshop, let's write a program that uses some of the concepts we learned:

Let's write a little program which asks for input on pizza toppings, puts those toppings in a list, then prints out the list at the end of the program.

# Summary



Let's review the programming concepts we learned in this workshop:

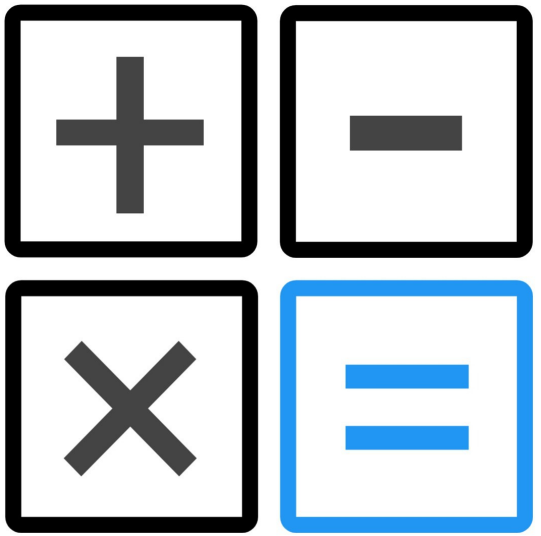
# Lists

```
>>> vegetables = ["carrots", "lettuce", "beets"]
>>> print(vegetables[0])
carrots
```

We learned about lists, which are variables with one or more items associated with them. Items in the list can be referenced by a process called **list indexing**.

# Basic Operators

And the other thing we learned in this workshop is Python basic operators, which are used to perform arithmetic operations.



# What's Next?

In the next HackerFrogs Afterschool programming workshop, we'll continue learning Python with the learnpython.org website.

# Next Workshop's Topics

- C-style string formatting

- String methods

- String splicing

# Extra Credit

Looking for more study material on this workshop's topics?

See this video's description for links to supplemental documents and exercises!

# Until Next Time, HackerFrogs!