🔙 to cover

Unit 2

Unit 2: Number Calculations & Data

Openers & Ideas

🔙 to divider

Mathematical Operators:

- Python provides the capability to perform basic arithmetic operations such as addition and multiplication

Parenthesis	(.....)	Exponents	**
Multiplication	*	Division	/
Addition	+	Subtraction	-

Advanced Operations: For more operations, you must import the math library using: “import math”

Square Root	math.sqrt(x)	**Trig**	math.sin(x)
Abs Value	math.fabs(x)	**Degrees**	math.degrees()
Log	math.log(x,base)	**Pi**	math.pi

# is Single Line Comment
Between ‘’’ and ‘’’ is a multiple line comment
—--------------------------

Input:

```
x=4
y=2
z=9
sol = z+x*y
print("The answer is " + sol)
```

Output:

```
Line 5: TypeError: cannot concatenate 'str' and
'int' objects
```

Input:

```
x=4
y=2
z=9
sol = z+x*y
print("The answer is " + str(sol))
```

Output:

```
The answer is 17
```

Unit 2 Review

🔙 to divider

*In 1821, at the age of 30, English mathematician Charles Babbage invented the Difference Engine to compile mathematical tables. As soon as it was completed, he had an idea to make it better by adding the ability to handle any type of calculation. Ada Lovelace created the first computer program and realized that Babbage's engine could be used for more than numeric calculations. Many see Babbage and Lovelace's work as the precursor to today's computers and computer programs.

- Charles Babbage conceived the general programming machine (Difference Engine and Analytical Machine)
- Alan Turing described a general purpose computer that could compute any problem
- Konrad Zuse built a program controlled computer (Souza Z3)
- John Mauchly and Presper Eckert built the ENIAC, the first all-electronic computer controlled by a program
- Manchester University built a prototype of a stored program computer
- The first computer ran a program from memory in 1948
- Multiple people built the computer overall

Function	**Description**	**Example**	**Output**

`sqrt(x)`	Takes the square root of x and returns a float	`import math`
		`print(math.sqrt(81))`
			9.0

`fabs(x)`	Takes the absolute value of x and returns it as a float	`import math`
		`print(math.fabs(-9))`
			9.0

`pow(x, y)`	Raises x to the y power and returns it as a float	`import math`
		`print(math.pow(3, 2))`
			9.0

Import random

Function	**Sample Usage**	**Description**

randint(x, y)	`import random`
	`x = random.randint(5, 10)`
		Returns a random integer between x and y, inclusive.

random()	`import random`
	`x = random.random()`
		Returns a random number from [0.0, 1.0) (i.e., greater than or equal to 0.0 but less than 1.0).

choice(sequence)	`import random`
	`animal = random.choice([“cat”, “dog”, “fish”, “snake”])`
		Picks a random element of a sequence. (In this example, animal could be randomly assigned to be cat, dog, fish, or snake). The sequence must always be contained within brackets.

Unit 2 Review

🔙 to divider

*A megabyte is one million bytes, a gigabyte is one thousand megabytes, a terabyte is one thousand gigabytes and a petabyte is one thousand terabytes. In bytes, a petabyte is 1,000,000,000,000,000 bytes.

max() function: accepts a list of values and returns the largest
min() function: accepts a list of values and returns the smallest

Ex: print(min("cat", "dog", "iguana", "anteater", "fish", "aardvark"))
Result: aardvark
Uses alphabetical order to compare strings

To plot data, use (import simpleplot)
*List data types can hold more than one value at a time
*List data types are signifies with square brackets and each value is separated by a comma
Exmple:

```
import simpleplot

dataset1 = [(1, 4), (1, 5), (2, 7), (4, 9)]
dataset2 = [(1, 2), (2, 7), (2, 5), (7, 6)]

simpleplot.plot_lines("Sample", 400, 300, "x", "y", [dataset1, dataset2], True, ["Dataset 1", "Dataset 2"])
```

The first parameter ("Sample") is the title. The second and third are the width (400) and height (300) of the graph. The fourth and fifth label the x and y axes. The next parameter contains our x and y values. The last two are optional. The True in this example says that we want to indicate the points on our graph and the last parameter gives a legend for the graph.

Different data types take up different amounts of space in memory and Python needs to know how much space to set aside for each variable.