```
1 import os
2
3 current_dir = os.getcwd()
4 print(current_dir)
```

```
/content
```

## ECL404 Skill Base Lab

## Module 1.1

# 1.1-Identifiers

The identifier is a name used to identify a variable, function, class, module, etc. The identifier is a combination of character digits and underscore. The identifier should start with a character or Underscore then use digit. The characters are A-Z or a-z,a UnderScore ( _ ) and digit (0-9). we should not use special characters ( #, @, $, %, ! ) in identifiers.

```
1 # Some valid variable names
2 var_1= 80  # integer
3 _var_2= 'a'    # string
4 SRK2021= 3.14
```

```
1  # some Invalid variable names
2  #1_var = 80
3  # OR
4  @var_2= 'a'
```

# 1.2-Keywords

Python Keywords are special reserved words that convey a special meaning to the compiler/interpreter. Each keyword has a special meaning and a specific operation. These keywords can't be used as a variable.

The keywords are some predefined and reserved words in python that have special meaning. Keywords are used to define the syntax of the coding. The keyword cannot be used as an identifier, function, and variable name. All the keywords in python are written in lower case expect True and False. There are 33 keywords in Python 3.7 let's go through all of them one by one. good reference

# 1.3- Comments

Comments are lines that exist in computer programs that are ignored by compilers and interpreters. Including comments in programs makes code more readable for humans as it provides some information or explanation about what each part of a program is doing.

Depending on the purpose of your program, comments can serve as notes to yourself or reminders, or they can be written with the intention of other programmers being able to understand what your code is doing.

In general, it is a good idea to write comments while you are writing or updating a program as it is easy to forget your thought process later on, and comments written later may be less useful in the long term.

## Comment Syntax

Comments in Python begin with a hash mark (#) and whitespace character and continue to the end of the line.

Generally, comments will look something like this:

```
1 #  This is a COMMENT and will be ignored in the CODE
```

# 1.4- Indentation

## What is Python Indentation

Compilers/interpreters generally do not know the sequence in which they need to execute the statements in a piece of code. Hence, to make it easier, we divide the code into several blocks of code and indent it. This indentation helps them understand the order in which each block/statement should be executed.

Similarly, Python indentation is a way of telling the Python interpreter that a series of statements belong to a particular block of code. In languages like C, C++, Java, we use curly braces { } to indicate the start and end of a block of code. In Python, we use space/tab as indentation to indicate the same to the compiler.

In simple, all the statements with the same distance (space) to the right, belong to the same block. In other words, statements belonging to a block will necessarily start from the same vertical line.

```
a = input('Enter the easiest programming language: ') #statement 1
if a == 'python': #statement 2
  print('Yes! You are right') #statement 3
else: #statement 4
  print('Nope! You are wrong') #statement 5
```

## 1.5- Variables ( Local and Global)

Easier to understand AFTER learning about FUNCTIONS SKIP this CODE block for now

```
1
```

## 1.6- Data types

### Built-in Data types

https://github.com/AnandKhandekar/Python-course-ECL404/blob/main/02a%20-%20Variables.md

Python has the following data types built-in by default, in these categories

Text type : str Numeric Types : int, float, complex Sequence Types: list, tuple, range Mapping Type : dict Bollean Types : bool Set Types

## Getting the Data Types

https://github.com/AnandKhandekar/Python-course-ECL404/blob/main/03%20-%20Input%20methods.md

We can CHECK which Data type is being used internally by using the type() function

```
1 x= "Hello world"
2 type(x)
```

```
1 x= 20
2 type(x)
```

```
1 x= 3.14
2 type(x)
```

```
1 x= 12.6j
2 type(x)
```

```
1 x= ["apple","banana","mango"]
2 type(x)
```

```
1 x= ("apple","banana","mango")
2 type(x)
```

```
1 x= range(6)
2 type(x)
```

```
1 x= {"name": "Anand Khandekar", "age" : 49 }  # eexplain the concept of KEY and VALUE
2 type(x)
```

dict

```
x =
type(x)
```

```
x = frozenset({"apple","banana","mango"})
type(x)
```

```
x = True
type(x)
```

## Setting the Specific Data type we want

To do so , w can use the pre defined constructor functions in Python

```
x = str("Hello World")
type(x)
```

```
x = int(20)
type(x)
```

```
x = float(22.189)
type(x)
```

and so on for all DATA types mentioned above

## 1.7- Arithmetic Operations

https://github.com/AnandKhandekar/Python-course-ECL404/blob/main/05%20-%20Operators.md

1

## 1.8- Comparative Operations

https://github.com/AnandKhandekar/Python-course-ECL404/blob/main/06%20-%20Boolean%20and%20Logical%20operators.md

1

## 1.9- Logical and Identity Operators

1

## 2.0- Bitwise Operators

1

## 2.1- Expressions

1

## 2.2 - Print Statements and Formats

1

## 2.3 - Input Statements and Formats

https://github.com/programiz/python-course/blob/master/03-input.md

```
1
```