My Intro....

- Name : Sameer Negi
- Occupation : Software Developer
- Age : 28
- Current Company : ESR Labs

- Github : https://github.com/sameernegi17/PythonBasic
- Email : sameer.negi17@gmail.com

Share This stream with your friends…….

Python..... & let's install pycharm

Time for Video

https://github.com/sameernegi17/PythonBasic/blob/main/Installation.mp4

Why Python ??

- Easy
- Python is a popular programming language.
- Easy to find Jobs. Can we used as:
  - Backend Language
  - Web Applications
  - Scripting
  - ML/DL

First STATEMENT : PRINT

```
Print(“My First Program”)
```

What we will cover today

1. Lot’s of Practical. I have 25 problems for you.
2. Data Types
3. Operators
4. Casting
5. If-Else
6. Functions

Not More Than that. I don’t want to scare you off

Taking Inputs From Users

INPUT STATEMENT

input(“write anything you want”)

.format() -> for formatting

```
anystring = “you are {}”

anystring.format(“strong”)
```

CREATING VARIABLE

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume). Rules for Python variables:

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (age, Age and AGE are three different variables)

Assign multiple values to variable

```
x, y, z = "Orange", "Banana", "Cherry"

print(x)

print(y)

print(z)
```

DATA TYPES

Built-in Data Types

Text Type: **`str`**
Numeric Types: **`int`**, **`float`**, `complex`
Sequence Types: **`list`**, **`tuple`**, **`range`**
Mapping Type: **`dict`**
Set Types: **`set`**, `frozenset`
Boolean Type: **`bool`**
Binary Types: `bytes`, `bytearray`, `memoryview`

type()

```
x = 5

print(type(x))
```

```
OUTPUT : <class 'int'>
```

CASTING

- There may be times when you want to specify a type on to a variable. This can be done with casting.
- `int()` - constructs an integer number from an integer literal, a float literal (by removing all decimals), or a string literal (providing the string represents a whole number)
- `float()` - constructs a float number from an integer literal, a float literal or a string literal (providing the string represents a float or an integer)
- `str()` - constructs a string from a wide variety of data types, including strings, integer literals and float literals

Operators

Python Arithmetic Operators

Operator	Name
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
**	Exponentiation
//	Floor Division

Python Assignment Operators

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

Python Comparison Operators

==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Python Logical Operators

and	Returns True if both statements are true	x < 5 and x < 10
or	Returns True if one of the statements is true	x < 5 or x < 4
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)

Python if-Else

Python Conditions and If statements

Python supports the usual logical conditions from mathematics:

- Equals: `a == b`
- Not Equals: `a != b`
- Less than: `a < b`
- Less than or equal to: `a <= b`
- Greater than: `a > b`
- Greater than or equal to: `a >= b`

These conditions can be used in several ways, most commonly in "if statements" and loops.

An "if statement" is written by using the `if` keyword.

Programs

1. To find a number is odd and even
2. Find the max of 3 numbers

Python functions

Defination

A function is a block of code which only runs when it is called.

You can pass data, known as parameters, into a function.

A function can return data as a result.

```
def my_function():

  print("Hello from a function")
```

Arguments

Information can be passed into functions as arguments.

Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

The following example has a function with one argument (fname). When the function is called, we pass along a first name, which is used inside the function to print the full name:

Create a calculator