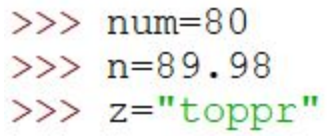
002: Variables and Datatypes

Learning Outcomes:	Concept of variable, rules of naming a variable, data types: int, float, String; Relational Operators

Definitions/Concepts	
Variable	• It is a unique memory location in the computer which holds the value we assign to it. eg. *a=1000* *here a is the variable which stores the value 1000 in it.*
Rules of naming a variable	• It can be a combination of letters and numbers, but should start with a letter not a number. *eg. account1, q3, num4, t6h* • The only special character that can be used is underscore (_) *eg. avg_score, t_5* • It can't have spaces in between • Keywords can't be used as variable names.
Keywords	• Words which convey special meaning in a programming language are called keywords. *e.g.* ***print*** *can't be used as variable*
Datatype	• The type of value stored in a variable is called the datatype of the variable.

Data Types		
int	It stands for integers.	*e.g. a=1000* *Here the data type of a is int*
float	It stands for floating point or decimal numbers.	*e.g. b=22.6* *Here, the data type of b is float*

String	Anything written within quotes(single or double) is of String Data type.	*e.g. c="toppr"* *Here, the data type of c is String*

Creating and assigning values to Variables	
• Unlike other languages, we don't need to specify the data type of variable and declare it. • We can simply create variables by assigning values to it.	
	Their data type are as follows: • num: int • n: float • z: String

Relational Operators	
• The operators used to compare two values are called Relational operators.	
• **>**	Greater than
• **<**	Less than
• **>=**	Greater than or equal to
• **<=**	Smaller than or equal to
• **==**	Equals to
• **!=**	Not equal to

Difference between = and ==	
The **single equal to** operator (**=**) is the **assignment operator**, which is used to give values to a variable.	The **double equal to** operator (**==**) is the **relational operator**, which is used to compare two things.

Operation on Variables	
• We can use the **print()** function to display the value of the variable on screen also.	

`>>> num=80`
`>>> n=89.98`
`>>> z="toppr"`
	`>>> print(num)`
	`80`
	`>>> print(n)`
	`89.98`
	`>>> print(z)`
	`toppr`

• We can use mathematical operators on variables and also assign those to another variable.	

`>>> n1=10`
`>>> n2=12`
`>>> n1+n2`
`22`
	`>>> sum=n1+n2`
	`>>> print(sum)`
	`22`
	`>>>`

Activity links and Solutions	
Student Activity 1: Variables	
#Activity 1: Create 2 variables storing integers.	`var1=6` `var2=1000`
#Activity 2: Create 2 variables storing decimal numbers.	`var3=8.3` `var4=0.44`
#Activity 3: Create 2 variables storing Strings.	`var5="toppr"` `var6='codr'`

#Activity 4:
Try to create variables with these name: var, 3top, new_4, n.k, for

```
var=5
new_4=90
```

These variables will be created easily as the variable names are valid.

```
3top=89
n.k=45
for=90
```

These variables will not be created as the variable names are not valid.

Student Activity 2: Operations on Variables

#Activity 1: Declare two variables with numbers(one integer and other float) and print their sum, difference, product, with proper message.

```
a=10
b=3
s=a+b
d=a-b
p=a*b
print("Sum of",a,"and",b,"is",s)
print("Difference of",a,"and",b,"is",d)
print("Product of",a,"and",b,"is",p)
```

#Activity 2: Divide the larger number by the smaller number and display the quotient, remainder and result of division.

```
quotient=a//b
remainder=a%b
result_of_division=a/b
print("Quotient=",quotient)
print("Remainder=",remainder)
print("Result of division=",
result_of_division)
```

#Activity 2: Divide the smaller number by the larger number and display the quotient, remainder and result of division.

```
quotient=b//a
remainder=b%a
result_of_division=b/a
print("Quotient=",quotient)
print("Remainder=",remainder)
print("Result of division=",
result_of_division)
```

Fun Fact
Google Search and YouTube are products that are powered by Python.