TX00FL42-3001

VARIABLES AND INTERACTIVE PROGRAMS

MUATH OTHMAN

LEARNING OBJECTIVES

- Understanding variables and their role in programming.
- Learning how to interact with users using input and output functions.
- Writing simple interactive Python programs.

INTRODUCTION TO PRINTING

		Built-in Functions		
abs()	divmod()	input()	open()	staticmethod()
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	__import__()
complex()	hash()	min()	set()	
delattr()	help()	next()	setattr()	
dict()	hex()	object()	slice()	
dir()	id()	oct()	sorted()	


print('Hello, world!')


print("Hello, world!")


print('"Hello", said Joe')

```
print("Good")
print("morning")
```


print("Good\nmorning")

INTRODUCTION TO VARIABLES

LET’S TAKE EXAMPLE

AA018465

INTERNATIONAL PASSPORT

Surname
MUATH

Given Names
OTHMAN

Date of birth
20-11-1995

Date of issue
01-09-2024

Date of expiry
01-09-2028

Passport No.
AA018465

Personal No.
012684529

Sex
M

Holder's signature
muath Othman

P<<Muath<<Othman<<<<<<<<<<<<<<<<<<<<<<<<

AA018465<<DDMMYYYY<<<<<<<<<<<<<<<<<<<<<<

AA051648

INTERNATIONAL PASSPORT

Surname
Sara

Given Names
SYDNEY

Date of birth
11-01-1990

Date of issue
12-02-2021

Date of expiry
12-02-2025

Passport No.
AA051648

Personal No.
512694358

Sex
F

Holder's signature
Sydney

P<<SARA<<SYDNEY<<<<<<<<<<<<<<<<<<<<<<<<<<<<

AA051648<<DDMMYYYY<<<<<<<<<<<<<<<<<<<<<<<<

```
surname = "Muath"
given_names = "Othman"
date_of_birth = "20.11.1995"
date_of_issue = "01.09.2024"
date_of_expiry = "01.09.2028"
passport_number = "AA018465"
personal_no = "012684529"
sex = "M"
```

LIFE WITHOUT VARIABLES


```
"Muath"
"Othman"
"20.11.1995"
"01.09.2024"
"01.09.2028"
"AA018465"
"012684529"
"M"
```

Exercises

VARIABLE NAIMING

1. ALLOWED CHARACTERS

- Letters (a-z, A-Z)
- Digits (0-9)
- Underscores (_)

1. ALLOWED CHARACTERS

```
valid_variable = 1
anotherVariable = 2
variable_3 = 3
_private_var = 4
```

2. CASE SENSITIVITY

```
name = "Alice"
Name = "Bob"
NAME = "Charlie"
```

3. RESERVED KEYWORDS

```
if = 10       # SyntaxError
class = "A"   # SyntaxError
```

Python has a set of reserved keywords that cannot be used as variable names

Let’s Check them together!

3. RESERVED KEYWORDS

```
import keyword
print(keyword.kwlist)
```

4. NAMING CONVENTIONS

```
my_variable = 5             # Snake_case (common for variable names)
MyClassName = "example"     # CamelCase (common for class names)
PI = 3.14159                # Constant (all uppercase)
```

5. AVOID SPECIAL CHARACTERS AND SPACES

```
# Invalid examples:
my-variable = 1       # SyntaxError
variable name = 2     # SyntaxError
```

6. DO NOT START WITH NUMBERS

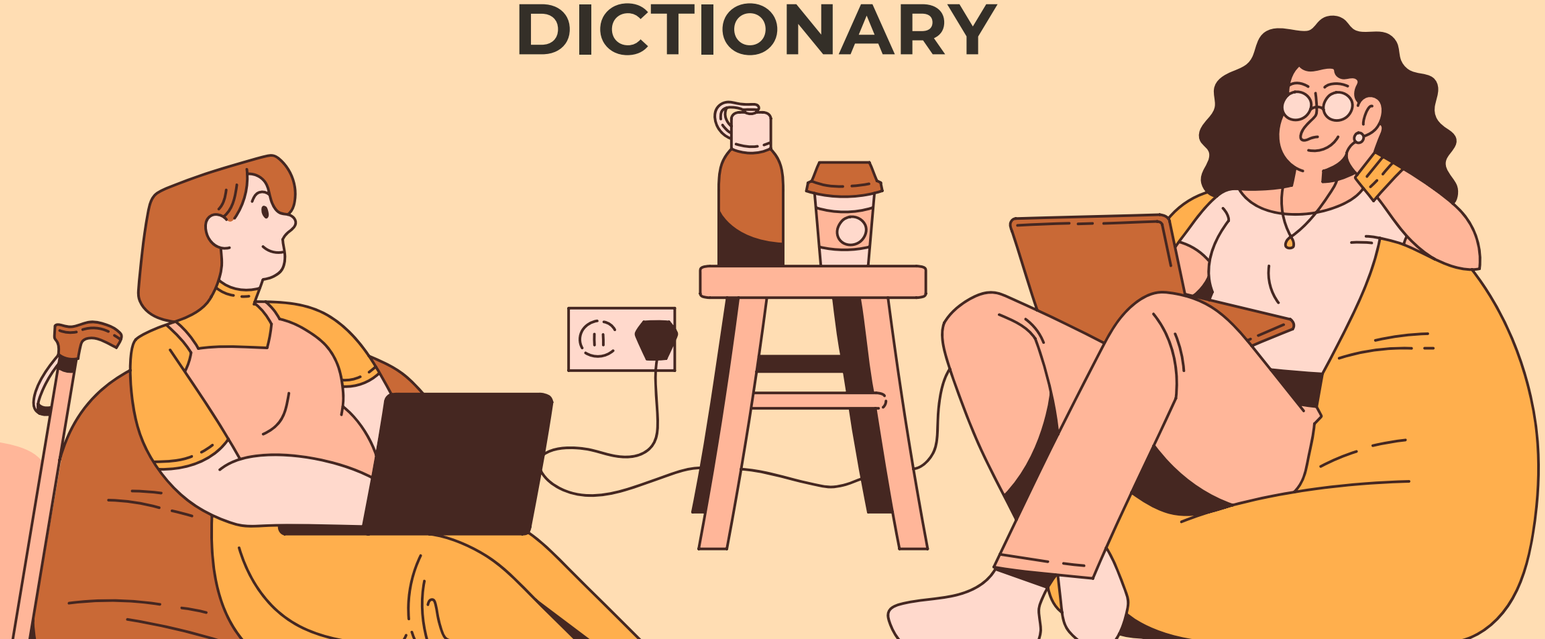
```
1variable = 10  # SyntaxError
```

Exercises

INTRODUCTION TO VARIABLE TYPES

COMMON TYPES

STRING
NUMBER
BOOLEAN
LIST
TUPLE
DICTIONARY



PYTHON'S NUMBER TYPE HAS FOUR SUB-TYPES

INTEGER: E.G., 4

LONG: E.G., 12756413000

FLOAT: E.G., 7.28 OR 4.0

COMPLEX: E.G., 3 - 2J

PYTHON'S NUMBER TYPE HAS FOUR SUB-TYPES

```
first = -9
second = 12_456_123_180
third = 4.973
fourth = -4 + 2j
```

YOU CAN ACCESS THE REAL AND IMAGINARY PARTS

```
print(fourth.real)
print(fourth.imag)
```

Exercises

INTRODUCTION TO INPUT

READING USER INPUT

```
input('Enter your name: ')
```

SAVING INPUT TO A VARIABLE

```
user_name = input('Enter your name: ')
```

TO CREATE A GREETING, COMBINE STRINGS WITH THE + OPERATOR

```
print("Nice to meet you, " + user + "!")
```

PYTHON PROVIDES A MORE CONCISE WAY TO FORMAT STRINGS USING F-STRINGS

```
print(f"Nice to meet you, {user}!")
```

COMPARING METHODS

```
print(f"Nice to meet you, {user}!")
```

```
print("Nice to meet you, " + user + "!")
```

WHAT IF I WANT NUMBER FROM USER?

		Built-in Functions		
abs()	divmod()	input()	open()	staticmethod()
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	__import__()
complex()	hash()	min()	set()	
delattr()	help()	next()	setattr()	
dict()	hex()	object()	slice()	
dir()	id()	oct()	sorted()	

MATHEMATICAL OPERATIONS

BASIC ARITHMETIC OPERATIONS

ADDITION: +

SUBTRACTION: -

MULTIPLICATION: *

DIVISION: /

ADVANCED ARITHMETIC OPERATIONS

MODULO OPERATOR: %

FLOOR DIVISION: //

EXPONENTIATION: **

DEFINING OPERATION ORDER

```
result = (2 + 3) * 4   # Output: 20
```

WHAT IS TYPE CONVERSION?

COMMON DATA TYPES INCLUDE

STRING: STR

INTEGER: INT

FLOAT: FLOAT

USE INT() TO CONVERT A STRING TO AN INTEGER

```
num = int("10")
```

USE FLOAT() TO CONVERT A STRING TO A FLOAT

```
num = int("10")
```

USE STR() TO CONVERT A NUMBER TO A STRING

```
text = "The result is " + str(100)
```

INTRODUCTION TO OUTPUT FORMATTING

TO ENSURE CELSIUS DEGREES ARE SHOWN WITH TWO DECIMALS

```
celsius = 123489.902343
print(f"The temperature in Celsius: {celsius:6.2f}")
```

COMMON DATA TYPES INCLUDE

.5F: FLOAT WITH 5 DECIMAL PLACES
10.2F: FLOAT WITH 2 DECIMALS IN A 10-CHARACTER WIDE FIELD
<20S: LEFT-JUSTIFIED STRING IN A 20-CHARACTER WIDE FIELD
8D: INTEGER IN AN 8-CHARACTER WIDE FIELD

CONVERTING FAHRENHEIT TO CELSIUS