Python Course Python overview

Feb 2023 Matin KarimPour



Let's write our first python code :))

Code overview print("your final budget is ", budget)

```
python:)
shoping_list = ["oil", "potato", "tomato", "rice"]
prices = {
    "oil": 70,
    "potato":15,
    "tomato": 20,
    "rice": 300
budget = 400
basket = []
for item in shoping_list:
    price = prices[item]
    if price < budget:
        basket.append(item)
        budget = budget - price
    else:
        print("budget is not enough for this ", item)
print("your final basket is ", basket)
```

Variable

```
python:)
shoping_list = ["oil", "potato", "tomato", "rice"]
prices = {
    "oil": 70,
    "potato":15,
    "tomato": 20,
    "rice": 300
budget = 400
basket = []
```

Integer



List

```
python:)
shoping_list = ["oil", "potato", "tomato", "rice"]
basket = []
```

Dictionary

```
pyth
prices = {
    "oil": 70,
    "potato":15,
    "tomato": 20,
    "rice": 300
```

Loop and Condition

```
python:)

for item in shoping_list:
    price = prices[item]
    if price < budget:
        basket.append(item)
        budget = budget - price
    else:
        print("budget is not enough for this ", item)return go(f, seed, [])
}</pre>
```

Print output



Variables, expressions, Statements

Values and types

```
python :)
print(10)
#10
type('Python Course')
# <class 'str'>
type(98)
# <class 'int'>
type(1.7)
# <class 'float'>
type('30')
# <class 'str'>
```

Variables

```
python:)
message = 'karim and his friends'
number = 17
pi = 3.1415926535897931
print(number)
#17
print(pi)
# 3.1415926535897931
type(message)
#<class 'str'>
```

Try these names

```
python:)

23pycourse = "amazing"
ye@r = 2023
class = "python"
```

These names are illegal

```
python:)

23pycourse = "amazing"
ye@r = 2023
class = "python"
# SyntaxError: invalid syntax
```

Python keywords

False	break	for	not
None	class	from	or
True	continue	global	pass
peg_parser	def	if	raise
and	del	import	return
as	elif	in	try
assert	else	is	while
async	except	lambda	with
await	finally	nonlocal	yield

Operators and operands

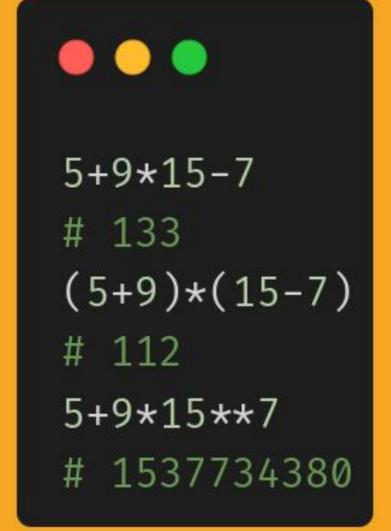
The operators + - * /
and ** perform addition,
subtraction,
multiplication, division,
and exponentiation, as
in the following
examples:

```
python :)
minute = 59
minute/60
# 0.9833333333333333
20 + 32
hour-1
hour * 60 + minute
5**2
(5+9)*(15-7)
```

Order of operations

PEMDAS

- **1** Parentheses
- 2. Exponentiation
- 3. Multiplication & Division
- 4. Addition & Subtraction



User input

When this function is called, the program stops and waits for the user to type something. When the user presses Return or Enter, the program resumes and input returns what the user typed as a string.

```
python:)
name = input('What is
your name?\n')
# What is your name?
# Matin
print(name)
# Matin
```

Any question?

Please scan this QR code and complete the survey.

