Beginners Workbook Solutions

DATA TYPES

Answer 1 : sect, decimal
Answer 2 : `if`
Answer 3 :

```
Float
30
```

```
25.0
```

Answer 4 :

float	-	45.05
complex	-	2 + 5y
bool	-	False
string	-	'python'
bytes	-	b'python'
intege	-	245
list	-	['Java', 'C++']
numeric string	-	'176'

VARIABLES

Answer 1

```python
a = 10
b = 20

print(a)
print(b)

a, b = b, a

print(a)
print(b)
```

Answer 2

```python
num = 1
print(num)
num += 10
print(num)
```

Answer 3

```
name_1 = "Jake"
name_2 = "David"
name_3 = "Tammy"

print("Hello " + name_1)
print("Hello " + name_2)
print("Hello " + name_3)
```

Answer 4:

Legal -

surname, encoded_var, VarString, IntVar, __init__, local_var

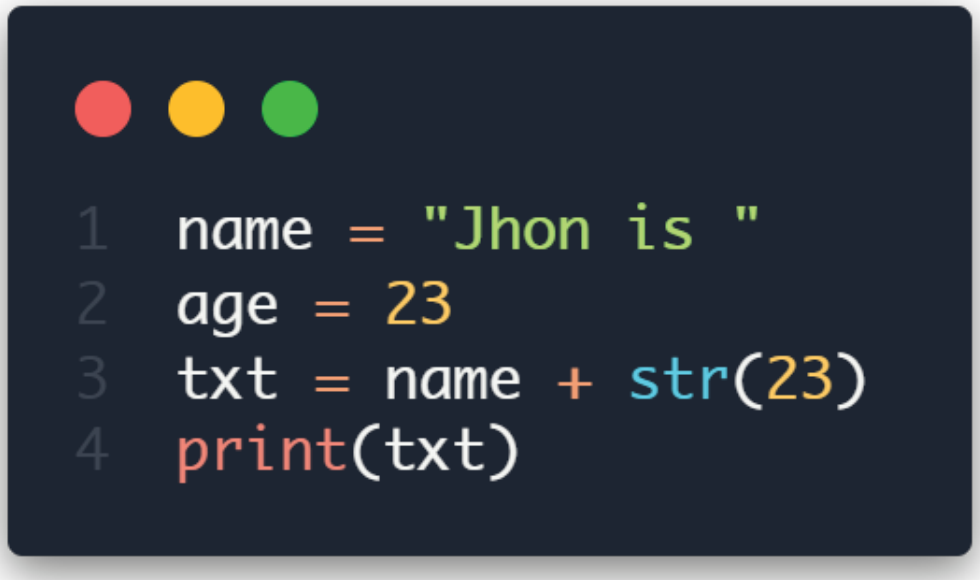
Illegal -

$money, <mood>, 2AddedVar, str, 6int9, Amt-A/c

Answer 5

```
first = "YoJohnSmith!"
second = 52591
```

Answer 6



```
name = "Jhon is "
age = 23
txt = name + str(23)
print(txt)
```

STRINGS

Answer 1

```
'It\'s Python'
'"Python" is easy'
"He said, 'GO' John"
'Well who\'s this'
```

Answer 2

```
quote = "the banna is yellow\
      but orange is orange"
print(
    quote.title()
)
```

Answer 3

```python
Str = 'Jhon Smith'
print(Str.lower())  # jhon smith
print(Str.upper())  # JHON SMITH
print(Str[0])  # J
print(Str[-2:])  # th
print(Str[5:])  # Smith
print(Str[1:4])  # hon
print(len(Str))  # 10
```

Answer 4

```python
code = chr(74) + chr(85) + \
       chr(83) + chr(84) + \
       chr(95) + chr(70) + \
       chr(85) + chr(78)

print(code)
```

Answer 5

```
-1
```

NUMBERS

Answer 1

```python
principal = 35000
rate = 3.5
time = 3

simple_interest = (principal * rate * time) / 100
amount = principal + simple_interest
print("Amount:", amount)
```

Answer 2

```python
a = 367
b = 255

print(a + b)
print(a - b)
print(a * b)
print(a / b)
print(a // b)
print(a % b)
```

Answer 3

```python
num = 45.495567
print(round(num, 5))
```

Answer 4

```python
basic_salary = 75000  # can be anything

gross_salary = (100 - 40 + 20) / basic_salary * 100
print(gross_salary)
```

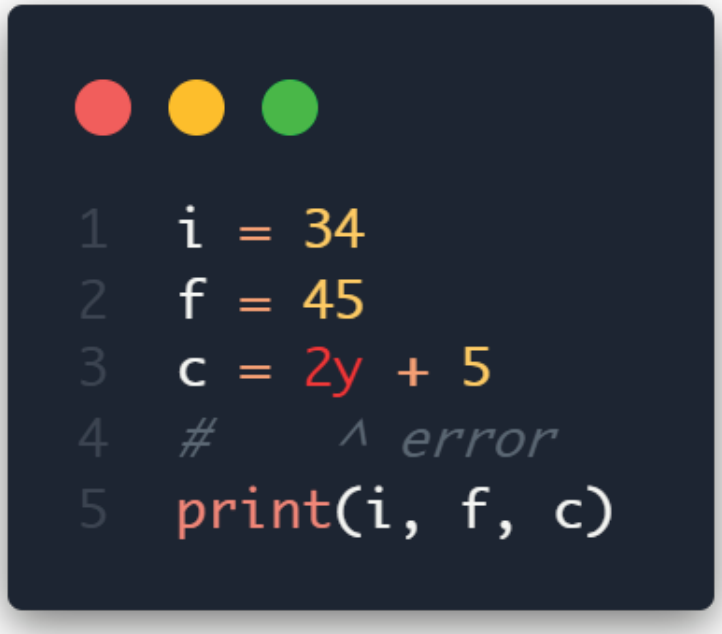
Answer 5

```python
# point 1
cmplx_num = 45 + 125j
print(cmplx_num.imag)

# point 2
num1 = 17.492720
print(round(num, 2))

```

```python
# point 3
number_lst = [123456, 123864, 123987, 123945]
print(max(number_lst))

# point 4
print(hex(34))

# point 5
print(pow(23, 11))

# point 6
num2 = 55.994521
print(round(num2))

# point 7
numbers_lst = [987456, 987864, 987987, 987945]
print(min(numbers_lst))

```

Answer 6

```python
a = 4j + 54
b = int(a)
# error ^
c = b + 6j + 2
print(c)
```



```
i = 34
f = 45
c = 2y + 5
#     ^ error
print(i, f, c)
```

Answer 7

```
7.19838019216856e+24
```

BOOLEANS

Answer 1

```
True
True
False
False
```

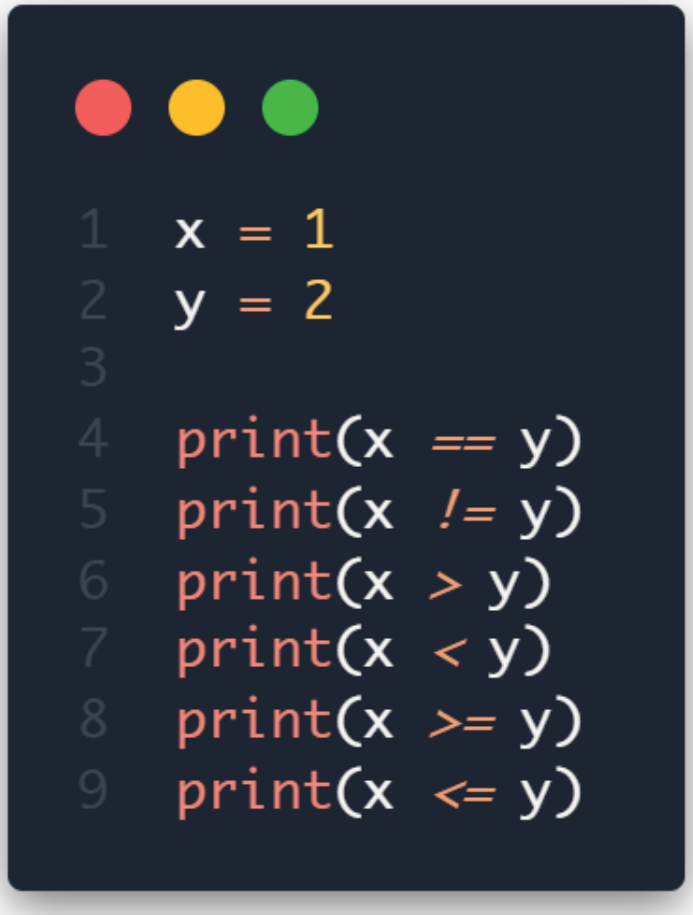
Answer 2

```
1  print(1 > 2)
2  print("n" in "abc")
3  print(bool(0))
```

Answer 3

```
1  print(1 == 2)
2  print("a" in "abc")
3  print(bool(1))
```

Answer 4



```
x = 1
y = 2

print(x == y)
print(x != y)
print(x > y)
print(x < y)
print(x >= y)
print(x <= y)
```

LISTS

Answer 1

```python
lang = ['C', 'C++', 'Java',
 'Python', 'Js']

lang.append('Ruby')
lang.remove('C')
lang[-2] = "JavaScript"
lang.sort()
print(lang)
```

Answer 2

```python
# 2
odd = [1, 3, 5, 7, 9]
eve = [2, 4, 6, 8]

num = odd + eve
# or num = [*odd, *eve]
```

```
print(len(num))

# reverse
num.sort()
num.reverse()
# or
# num = sorted(num, reverse=True)

num.clear()  # ;-P
```

Answer 3

```
lst = [1, 2, 3, 4, 5, 6, 7, 8,
    9, 10, 11, 12, 13, 14, 15, 16,
    17, 18, 19, 20, 21, 22, 23, 24, 25]
# or lst = list(range(1, 26))

# new list to store the output
new_lst = []
# adding 1 to 5, note the indexes
new_lst = lst[0:5]
# adding 9 to 12
new_lst += lst[8:13]
# removing 11
new_lst.remove(11)
```

```
13  new_lst.remove(11)
14  # adding 17 to 22
15  new_lst += lst[16:22]
16  # adding 24, 25
17  new_lst += [24, 25]
18
19  print(new_lst)
```

TUPLES

Answer 1

```python
tpl = (18, 4)
# unpacking with *
result = divmod(*tpl)
print(result)
```

Answer 2

```python
lst = [
    ("Cookies", 12.0),
    ("Biscuits", 6.0),
    ("Banna", 8.0),
    ("Apple", 7.25),
    ("Cake", 16.5),
]

def sorter(ele):
    return ele[1]
```

```
11
12  lst.sort(key=sorter)
13  lst.reverse()
14
15  print(lst)
```

The explanation of the above code can be found at: udemy.com/course/full-python-programming/learn/lecture/23264876#questions/15400508

Answer 3

```
1  lst = [(1, 2), (3, 5),
2         (), (12, ),
3         (11, 17), ()]
4  empty_tpl = ()
5
6  lst.remove(empty_tpl)
7  lst.remove(empty_tpl)
8  print(lst)
```

SETS

Answer 1

```
{33, 70, 40, 10, 44, 50, 51, 20, 60, 30}
{50, 10, 20, 30}
{40, 60, 70}
{33, 51, 44}
{33, 70, 40, 44, 51, 60}
False
False
```

Answer 2 :
Both **`discard()`** and **`remove()`** are used to remove elemets from a **`set`**, but if the passed element isn't present in the set the **`remove()`** raises error, while **`discard()`** ignores the error.

Answer 3 : **<=** or **>=**

Answer 4

Answer 5

```python
# Sets are unordered,
# i.e. they aren't indexed
# example :
st = {1, 2, 3, 5}
print(st[1])
```

Answer 6

```python
s = {10, 2, -3, 4, 5, 88}
# number of items in the set
print(len(s))
# max element in the set
print(max(s))
# min element in the set
print(min(s))
# sum of all elements in the set
print(sum(s))
# does set contains 77
print(77 in s)
# does set contains -3
print(-3 in s)
```

DICTIONARIES

Answer 1

```
students = ["Todd", "Evans",
            "Ravi", "Jhon"]
marks = [88, 76, 92, 87]

dt = {
    students[0]: marks[0],
    students[1]: marks[1],
    students[2]: marks[2],
    students[3]: marks[3]
}

```

Answer 2

```
str_lst = ["Jam", "Panda", "Bag", "Sky"]
dt = {str_lst[0]: len(str_lst[0]),
      str_lst[1]: len(str_lst[1]),
      str_lst[2]: len(str_lst[2]),
      str_lst[3]: len(str_lst[3])}
```

Answer 3

```
dt = {'A': 65, 'B': 66, 'C': 67}
dt1 = {'E': 68, 'F': 69}

dt.update(dt1)

dt['E'] = 69
dt['F'] = 70
```

Hopefully you've solved the problems correctly!
If you didn't understand any part just drop it in the Q&A Section.

- Rahul Mula