Python CS-521

Eugene Pinsky
Department of Computer Science
Metropolitan College, Boston University
Boston, MA 02215
email: epinsky@bu.edu

May 17, 2020

Abstract

This course will present an effective approach to help you learn Python. With extensive use of graphical illustrations, we will build understanding of Python and its capabilities by learning through many simple examples and analogies. The class will involve active student participation, discussions, and programming exercises. This approach will help you build a strong foundation in Python that you will be able to effectively apply in real-job situations and future courses.

EXERCISED

TUPLES

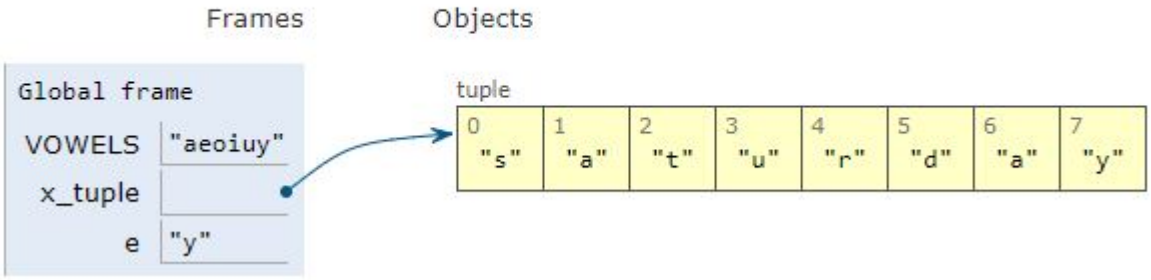
- print consonants in *x_tuple*:

```
x_tuple=tuple("saturday")
```

Solution:

```
VOWELS  = "aeoiuy"
x_tuple = tuple("saturday")
for e in x_tuple:
    if e not in VOWELS:
        print(e, end = " ")
```

```
s t r d
```



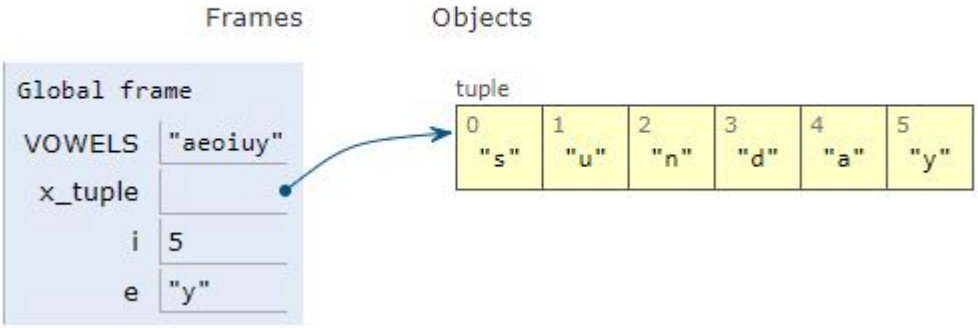
- print consonants and their positions in *x_tuple*:

```
x_tuple = tuple("sunday")
```

Solution:

```
VOWELS  = "aeoiuy"
x_tuple = tuple("sunday")
for i,e in enumerate(x_tuple):
    if e not in VOWELS:
        print((e,i), end = " ")
```

```
('s', 0) ('n', 2) ('d', 3)
```



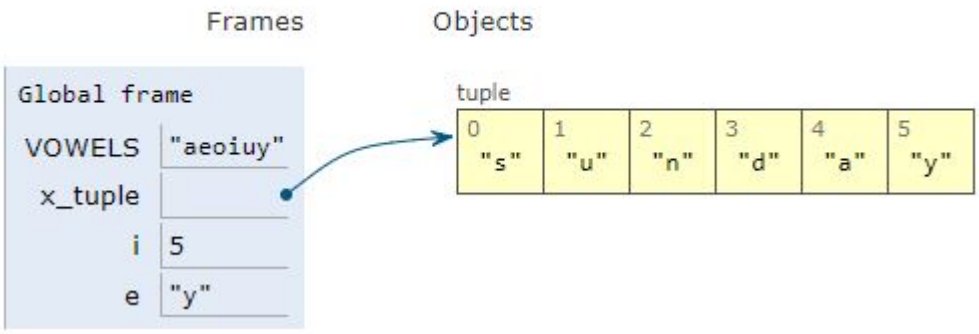
- write code without using *enumerate*()

Solution:

```
VOWELS  = "aeoiuy"
x_tuple = tuple("sunday")

for i in range(len(x_tuple)):
    e = x_tuple[i]
    if e not in VOWELS:
        print((e,i), end = " ")
```

```
('s', 0) ('n', 2) ('d', 3)
```



- compute elements from *x_tuple*:

```
x_tuple = tuple("tuesday")
```

(a) a = x_tuple[0]

(b) b = x_tuple[1]

(c) c = x_tuple[-1]

(d) d = x_tuple[5]

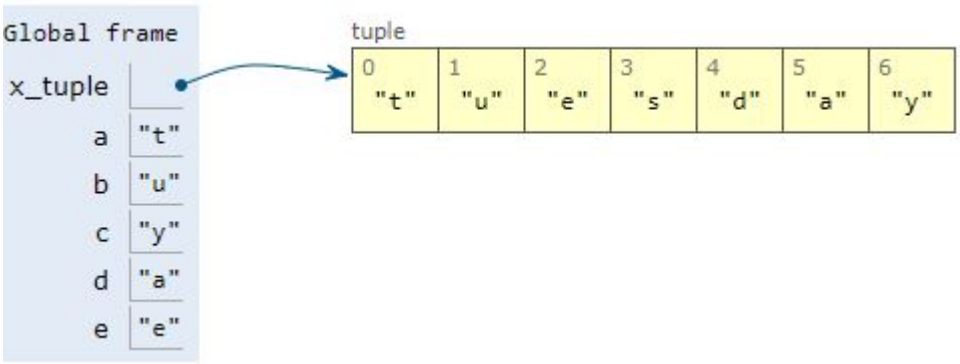
(e) e = x_tuple[-5]

(f) f = x_tuple[25]

Solution:

```
x_tuple = tuple("tuesday")

a = x_tuple[0]
b = x_tuple[1]
c = x_tuple[-1]
d = x_tuple[5]
e = x_tuple[-5]
f = x_tuple[25]     # error
```



- compute slices from *x_tuple*:

```
x_tuple=tuple("wednesday")
```

(a) a = x_tuple[0 : 10 : 2]

(b) b = x_tuple[1 : 9 : 3]

(c) c = x_tuple[-10 : -2 : 3]

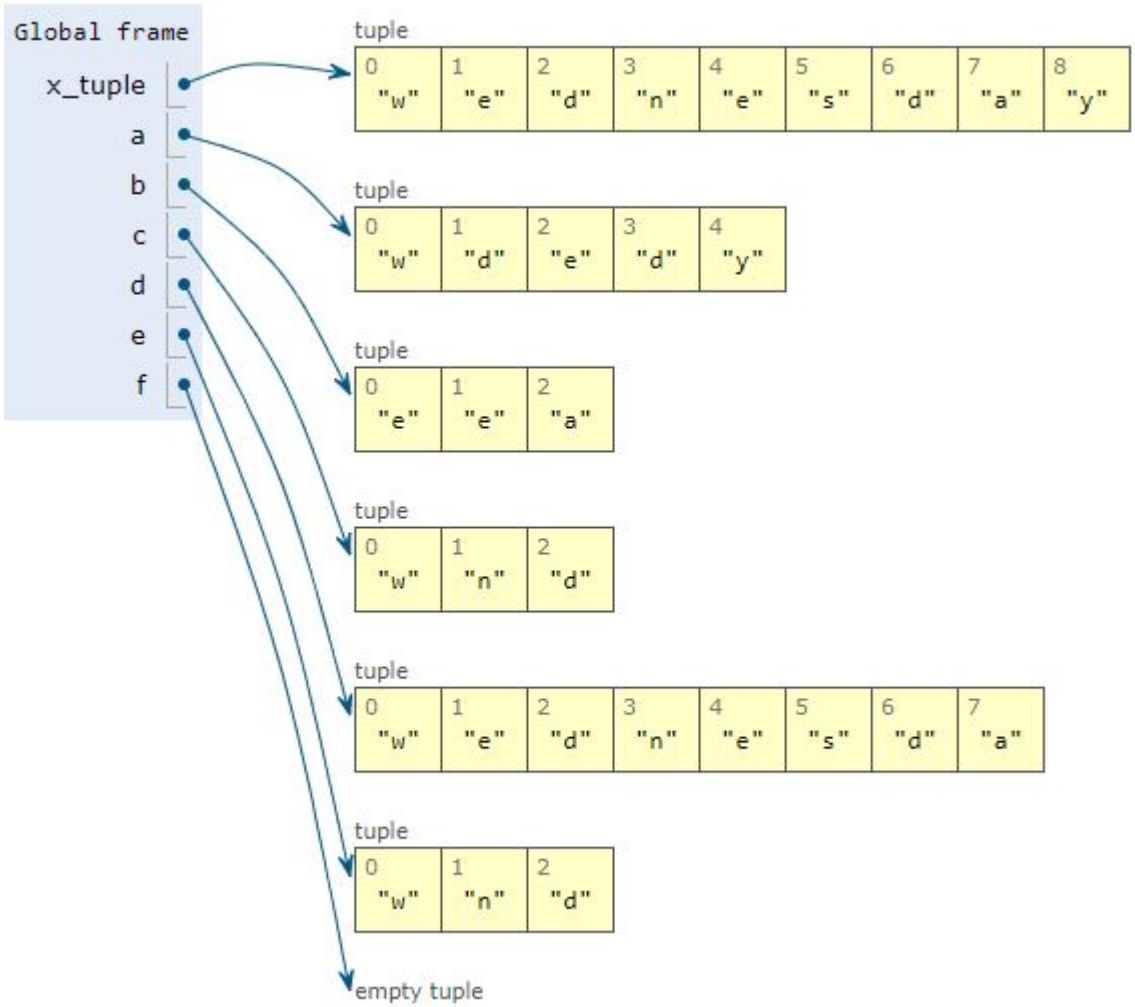
(d) d = x_tuple[0 : -1 : 1]

(e) e = x_tuple[0 : -2 : 3]

(f) f = x_tuple[30 : 5 : 5]

```
x_tuple = tuple("wednesday")

a = x_tuple[0   : 10 : 2]
b = x_tuple[1   : 9  : 3]
c = x_tuple[-10 : -2 : 3]
d = x_tuple[0   : -1 : 1]
e = x_tuple[0   : -2 : 3]
f = x_tuple[30  : 5  : 5]
```


Global frame
x_tuple
a
b
c
d
e
f
tuple
0 "w"
1 "e"
2 "d"
3 "n"
4 "e"
5 "s"
6 "d"
7 "a"
8 "y"
tuple
0 "w"
1 "d"
2 "e"
3 "d"
4 "y"
tuple
0 "e"
1 "e"
2 "a"
tuple
0 "w"
1 "n"
2 "d"
tuple
0 "w"
1 "e"
2 "d"
3 "n"
4 "e"
5 "s"
6 "d"
7 "a"
tuple
0 "w"
1 "n"
2 "d"
empty tuple

- compute slices from *x_tuple*:

```
x_tuple=tuple("thursday")
```

(a) a = x_tuple[10 : 0 : -1]

(b) b = x_tuple[10 : : -2]

(c) c = x_tuple[: : -2]

(d) d = x_tuple[: : -3]

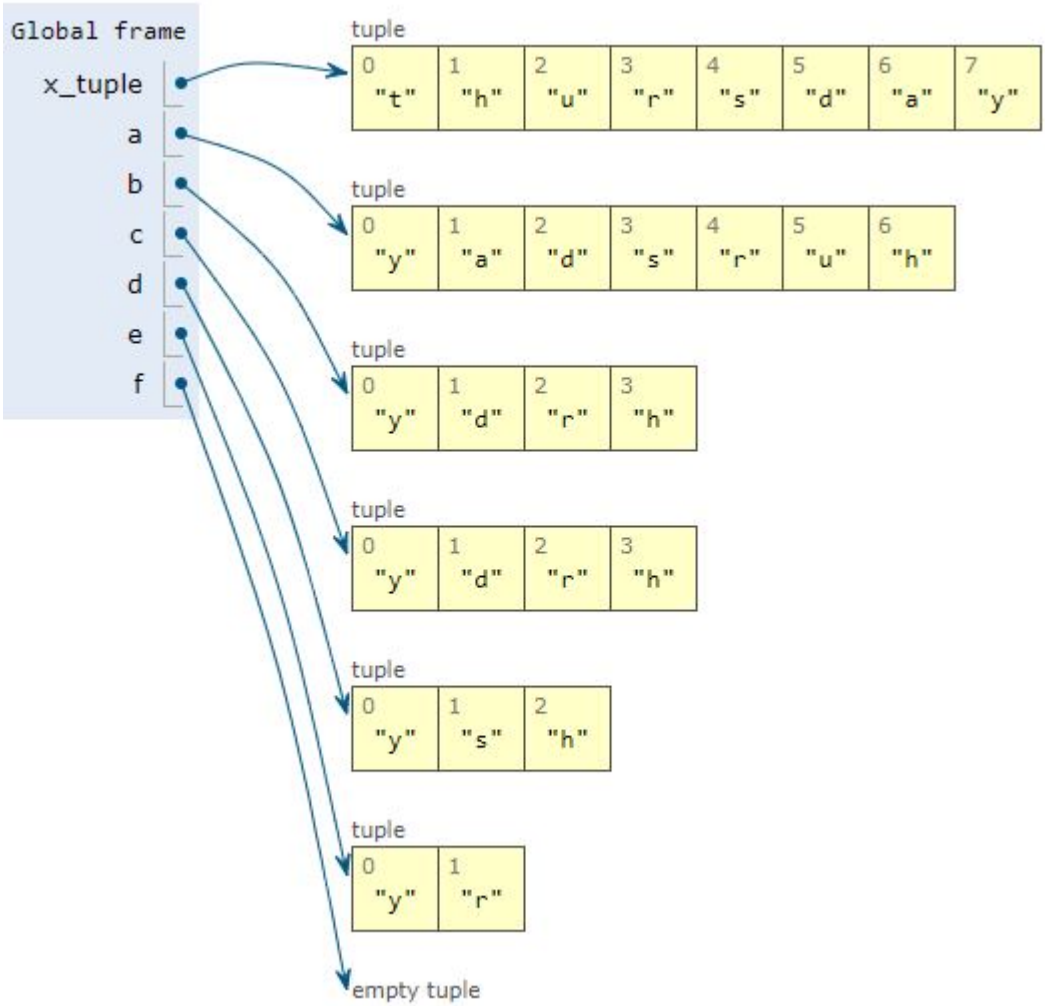
(e) e = x_tuple[: : -4]

(f) f = x_tuple[0 : -1 : -1]

Solution:

```
x_tuple = tuple("thursday")

a = x_tuple[10  : 0  : -1]
b = x_tuple[10  :    : -2]
c = x_tuple[    :    : -2]
d = x_tuple[    :    : -3]
e = x_tuple[    :    : -4]
f = x_tuple[0   : -1 : -1]
```


Global frame
x_tuple
a
b
c
d
e
f
tuple
0 "t"
1 "h"
2 "u"
3 "r"
4 "s"
5 "d"
6 "a"
7 "y"
tuple
0 "y"
1 "a"
2 "d"
3 "s"
4 "r"
5 "u"
6 "h"
tuple
0 "y"
1 "d"
2 "r"
3 "h"
tuple
0 "y"
1 "d"
2 "r"
3 "h"
tuple
0 "y"
1 "s"
2 "h"
tuple
0 "y"
1 "r"
empty tuple

- count consonants in *x_tuple*:

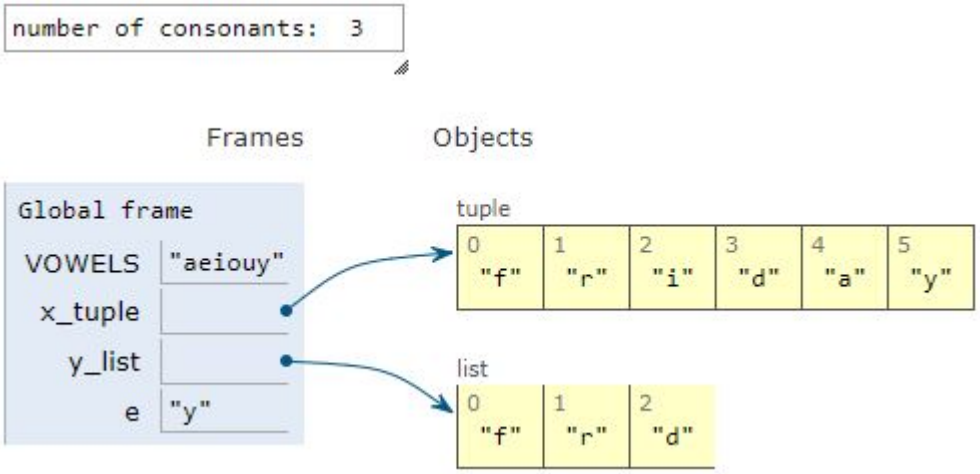
```
x_tuple = tuple("friday")
```

Solution:

```
VOWELS  = "aeiouy"
x_tuple = tuple("friday")
y_list  = []  # for distinct constants

for e in x_tuple:
    if e not in VOWELS and e not in y_list:
        y_list.append(e)

print("number of consonants: ", len(y_list))
```


number of consonants: 3
Frames
Objects
Global frame
VOWELS
"aeiouy"
x_tuple
y_list
e
"y"
tuple
0
"f"
1
"r"
2
"i"
3
"d"
4
"a"
5
"y"
list
0
"f"
1
"r"
2
"d"

- print indices for first occurence of consonants in *y_tuple*:

```
y_tuple = tuple("March")
```

Solution:

```
VOWELS  = "aeiouy"
x_tuple = tuple("friday")
y_list  = []  # for distinct constants

for e in x_tuple:
    if e not in VOWELS and e not in y_list:
        y_list.append(e)

for e in y_list:
    first  = x_tuple.index(e)
    print(e, "first index: ", first)
```

```
f first index:  0
r first index:  1
d first index:  3
```