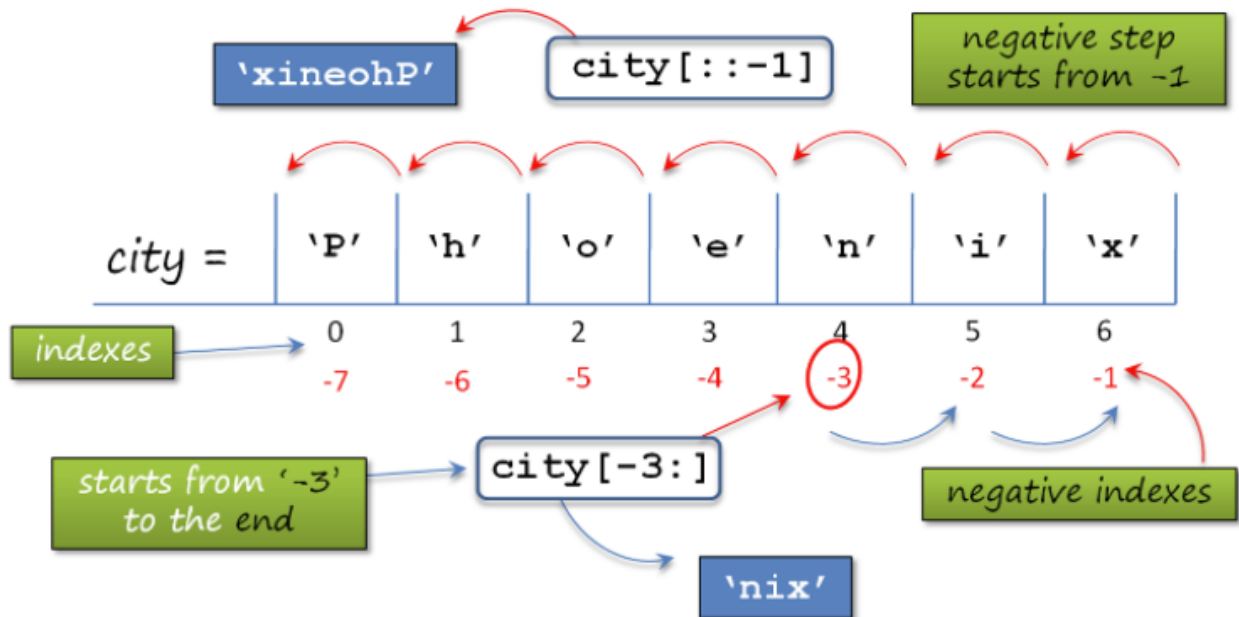


# 21.06.21 python

## iterable

Python'da gezinilebilen(iterable) nesne olarak kullanılabilen bir nesneye iterable obje denir. Bu temel olarak nesnedeki dizinin gezinilebileceği ve ilerlenebileceği anlamına gelir. Liste(list), sözlük(dict), kümeler(set) ve hatta range gibi Python koleksiyonlarının çoğu iterable olarak ele alınabilir



ucd Ini

## Indexing & Slicing Strings

Here is an example of *Pre-Class* content:

```
1 city = 'Phoenix'
2
3 print(city[1:]) # starts from index 1 to the end
4 print(city[:6]) # starts from zero to 5th index
5 print(city[:2]) # starts from zero to end by 2 step
6 print(city[1:2]) # starts from index 1 to the end by 2 step
7 print(city[-3:]) # starts from index -3 to the end
8 print(city[::-1]) # negative step starts from the end to zero
9
```

```
1 hoenix
2 Phoeni
3 Ponx
4 hei
5 nix
6 xineohP
7
```

```
fruit = 'Orange'
```

```
print('Word      : ', fruit)
```

```
print('First letter : ', fruit[0])
```

```
print('Second letter : ', fruit[1])
```

```
print("3rd to 5th letters : ", fruit[2:5])
```

```
print("Letter all after 3rd : ", fruit[2:])
```

Zoom

Presentation Session

app.peardeck.com/presentation/epkbnak/projector

Dataset list - A.S., AWS & DevOps M., python as module..., Clarusway-dev-7..., Python man funct..., Sessions - Pear D..., Kahoot! - Team Ka..., Python-program..., Pyformat: Using..., Learn JSON - Full..., Codingbat Python..., Codingbat Python..., Reading List

## ► Negative Indexing Strings

► Negative indexing works as the same:

step = positive

The diagram shows the string 'orange' with its characters mapped to indices. Positive indices (0-5) are shown below the string, and negative indices (-6 to -1) are shown below the positive ones. The characters 'r', 'a', and 'n' are circled in red, and their corresponding negative indices (-5, -4, -3) are also circled in red. A red arrow points from the word 'step' to the negative indices. To the right, a blue bracket indicates the slice [-5:-2], with arrows pointing to the corresponding indices in the diagram. Below this, another blue bracket indicates the slice [1:4], with arrows pointing to the corresponding indices in the diagram.

CLARUSWAY®  
WAY TO REINVENT YOURSELF

ucd Ini

```
a='orange'
```

```
a[-5:-2]
```

```
'ran'
```

```
'clarus'[-3::-1]
```

```
'ralc'
```

```
In [43]: hayvan = 'hippopotamus'|
```

```
In [45]: hayvan[-5:-2:1]
```

```
Out[45]: 'tam'
```

```
In [46]: hayvan[-5:-2:-1]
```

```
Out[46]: ''
```

```
In [42]: hayvan[-2:-5:-1]
```

```
Out[42]: 'uma'
```

String Formatting with Arithmetic Syntax

Another example :

```
str_one = 'upper'
str_two = 3 * 'upper'
str_comb = str_one * 3
print(str_two)
print(str_comb)
print(*str_one)
```

upperupperupper  
upperupperupper  
u p p e r

CLARUSWAY®  
WAY TO REINVENT YOURSELF

21

Zoom

Presentation Session

String Formatting with Arithmetic Syntax

► The output :

```
string_1 = 'I am angry...'
print(* string_1)
string_2 = '1453'
print(* string_2)
'joseph@clarusway.com' # Do not use variable
print(* 'joseph@clarusway.com')
```

```
I   a m   a n g r y . . .
1 4 5 3
j o s e p h @ c l a r u s w a y . c o m
```

CLARUSWAY®  
WAY TO REINVENT YOURSELF

24

Zoom

Presentation Session

String Formatting with `string.format()` Method

► The formula syntax

```
'string {} string {} string'.format(data1, data2)
```

CLARUSWAY®  
WAY TO REINVENT YOURSELF

29

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/dfd0d866-7c6c-49b6-8397-02b1dd77dc6b/04\\_PythonBasic\\_21-June-2021.ipynb](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/dfd0d866-7c6c-49b6-8397-02b1dd77dc6b/04_PythonBasic_21-June-2021.ipynb)