



STRINGS

SESSION 12

Strings

A string is a sequence of characters enclosed with quotes(single or double)

Eg. 'Welcome to CS GE'

S=' Hello World' Here S is a variable holding a string Hello World

Len function is used to find the length of a string

```
>>>len(S)
```

```
>>>11
```

Individual characters are accessed using a technique known as **Indexing**

Non negative Indices

0	1	2	3	4	5	6	7	8	9	10
H	E	L	L	O		W	O	R	L	D
-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Negative Indices

```
>>>S[0]  
>>>'H'
```



+ Code + Text



```
[1] S='HELLO WORLD'  
S[6]
```

'W'

```
[2] S[-1]
```

'D'

```
[3] S[2]
```

'L'

```
[4] i= len(S)-1  
S[i]
```

'D'

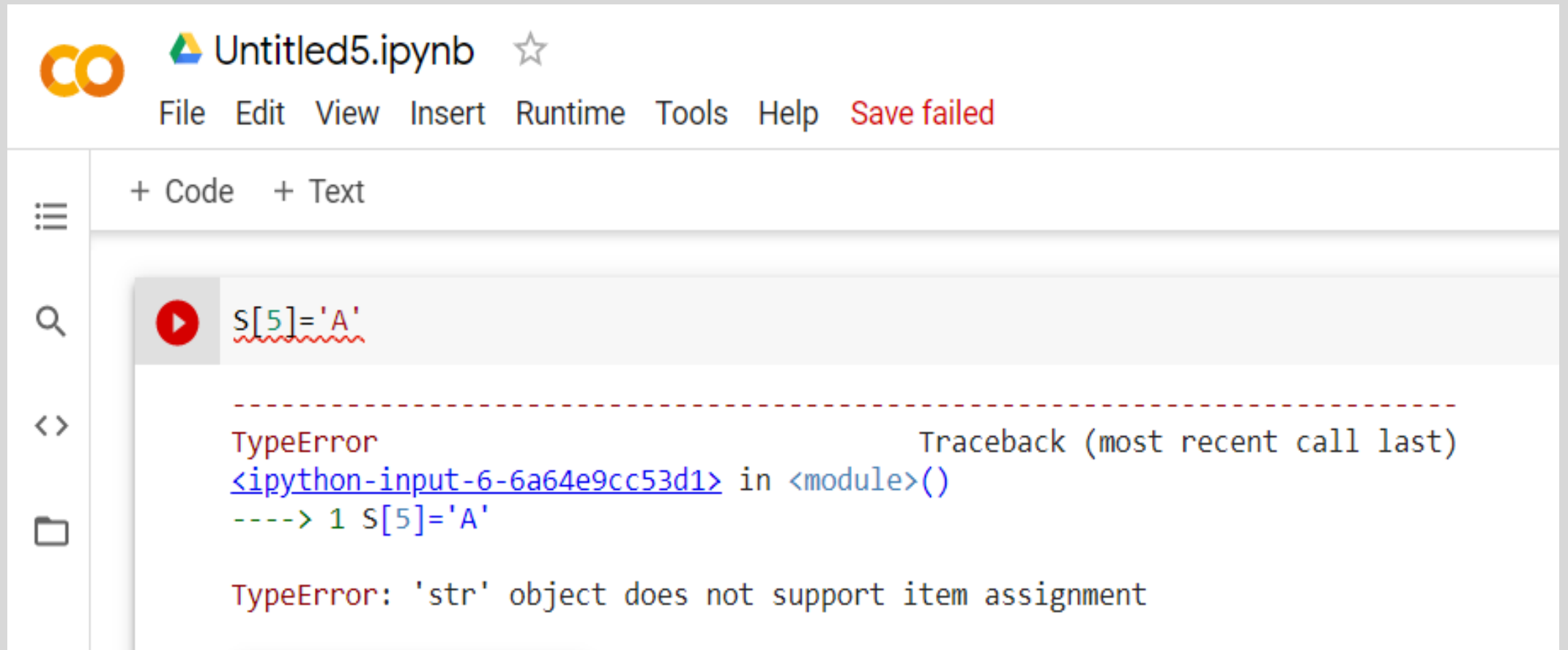


```
S[-i]
```

'E'

Strings are **immutable**

Components of a string cannot be altered any attempt to it will lead to error



The screenshot shows a Jupyter Notebook interface. At the top, there is a logo for Colab and the file name 'Untitled5.ipynb'. Below the file name is a menu bar with options: File, Edit, View, Insert, Runtime, Tools, Help, and a red 'Save failed' message. The notebook content area has a left sidebar with icons for a menu, search, expand/collapse, and a folder. The main area shows a code cell with a red play button icon and the code `S[5]='A'`. Below the code, a red dashed line separates the code from the error message. The error message is a `TypeError` with a traceback: `Traceback (most recent call last)`, `<ipython-input-6-6a64e9cc53d1> in <module>()`, and `----> 1 S[5]='A'`. The final line of the error is `TypeError: 'str' object does not support item assignment`.

```
co Untitled5.ipynb ☆
File Edit View Insert Runtime Tools Help Save failed

+ Code + Text

S[5]='A'

-----
TypeError                                Traceback (most recent call last)
<ipython-input-6-6a64e9cc53d1> in <module>()
----> 1 S[5]='A'

TypeError: 'str' object does not support item assignment
```

Strings can be concatenated using + operator

```
>>>'Computer' +' Science'
```

```
>>>'Computer Science'
```

```
>>>'Hi' +'How'+'are'+'you'
```

```
>>>'HiHowareyou'
```

```
>>>max('AZ', 'C', 'BD', 'BT')
```

```
>>>'C'
```

```
>>>min('BD', 'AZ', 'C')
```

```
>>>'AZ'
```

```
>>>max('hello', 'How', 'Are', 'You', 'sir')
```

```
>>>'sir'
```

Slicing- retrieving a substring

```
>>>message = 'HELLP WORLD'
```

0	1	2	3	4	5	6	7	8	9	10
H	E	L	L	O		W	O	R	L	D
-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
>>>message[0: 5]
```

```
>>>'HELLO'
```

```
>>>S[-10:-5]
```

```
>>>'ELLO '
```




+ Code + Text

```
[12] S[5:]
```

```
' WORLD'
```

```
[13] S[:]
```

```
'HELLO WORLD'
```

```
[16] S[:11]
```

```
'HELLO WORLD'
```

```
[17] S[:15]+S[15:]
```

```
'HELLO WORLD'
```



```
S[15:0]
```

```
..
```

Python also allows to extract a subsequence of the form start:end:inc

<>

```
[20] S='HELLO WORLD'  
S[0:len(S):2]
```

'HLOWRD'

```
[21] S[0:len(S):3]
```

'HLWL'



```
S[-1:-9:-2]
```



'DRWO'

Membership

- We can check membership of individual characters using in operator
- The expression would yield a True or a False

```
>>>'H' in 'Hello'
```

```
>>>True
```

```
>>>'H' in 'hello'
```

```
>>>False
```

Built in Functions in Strings

- **count()**- to find the number of occurrences of character in the string

```
>>>'hello how are are you'.count('o')
```

```
>>>3
```

```
S='hello how are are you'
```

S.count('o') will give 3

find()- returns the index of first occurrence of the substring (if found). If not found, it returns -1.

E.g.

```
quote = 'Let it be, let it be, let it be'
```

```
result = quote.find('let it')
```

```
print(result)
```

Answer = 11 (its small l in let it)

rfind()-returns the highest index (or rightmost index) of the substring (if found). If not found, it returns -1.

Eg

```
quote = 'Let it be, let it be, let it be'
```

```
result = quote.rfind('let it')
```

Answer -22

capitalize()-Converts the first character of the string to upper case

E.g.

```
txt = "hello, and welcome to my world."
```

```
x = txt.capitalize()
```

x will contain the string - Hello, and welcome to my world.

title()-returns a string with first letter of each word capitalized; a title cased string.

Str='hello how are you'

Str.title() will return

'Hello How Are You'

lower()-converts all uppercase characters in a string into lowercase characters and returns it.

Str='HELLO How ARE you'

Str.lower() will return

'hello how are you'

upper()-converts all lowercase characters in a string into uppercase characters and returns it.

Str='HELLO how ARE you'

Str.upper() will return

'HELLO HOW ARE YOU'

swapcase()- is used to covert lowercase letters in string to uppercase and vice versa

'Welcome to CS GE Class'.swapcase()

'wELCOME TO cs ge cLASS'

`islower()`-returns True if all alphabets in a string are lowercase alphabets. If the string contains at least one uppercase alphabet, it returns False.

Str= 'hello How are you'

Str.islower()-

Answer False (as H of how is in uppercase)

isupper()-returns whether or not all characters in a string are uppercased or not.

Str=' HOW ARE YOU'

Str.isupper() -returns true

'Hello how'.isupper()- returns false

istitle()-returns True if the string is a titlecased string. If not, it returns False.

That is only 1st character of every word is in capital rest all in lower case(the string should comprise of atleast one alphabet)

s = 'Python Is Good.'

print(s.istitle())

Answer True

A='123'.istitle() -it will return False

'Book 123'.istitle() - will return True

replace()-returns a copy of the string where all occurrences of a substring is replaced with another substring.

e.g

```
str='Hello how are you. Hello'
```

```
str.replace('Hello',' hola')
```

returns

```
-hola how are you.hola
```

split()-breaks up a string at the specified separator and returns a list of strings.

E.g

```
text= 'Love thy neighbor'
```

```
print(text.split())
```

```
output ['Love','thy','neighbor']
```

```
Colors='red, green, blue,pink'
```

```
Colors.split(',')-results in
```

```
['red',' green',' blue','pink']
```