Strings

What Is a String?

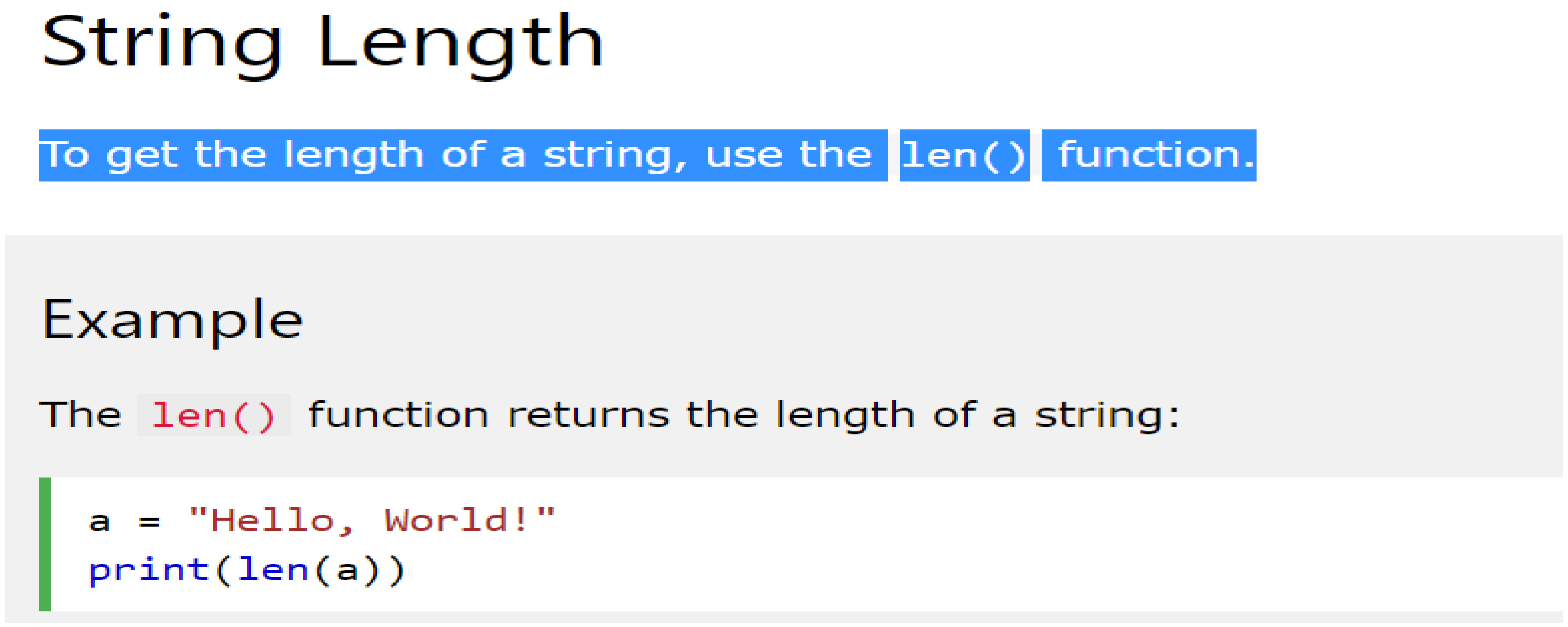
Strings are one of the fundamental Python data types. The term data type refers to what kind of data a value represents. Strings are used to represent text.

We say that strings are a fundamental data type because they can’t be broken down into smaller values of a different type.

Strings are “Immutable”, means once created they can not be changed.Whenever we try to modify an existing string new string will be created.

The string data type has a special abbreviated name in Python: str. You can see this by using type(), which is a function used to determine the data type of a given value.

- Python does not have a character data type, a single character is simply a string with a length of 1.
- Square brackets can be used to access elements of the string.
- a = "Hello, World!"
  print(a[1])
- OUTPUT-e
- Get the character at position 1 (remember that the first character has the position 0):



String Length

To get the length of a string, use the `len()` function.

Example

The `len()` function returns the length of a string:

```
a = "Hello, World!"
print(len(a))
```

Check String

To check if a certain phrase or character is present in a string, we can use the keyword `in`.

Example

Check if "free" is present in the following text:

```
txt = "The best things in life are free!"
print("free" in txt)
```

Use it in an `if` statement:

Example

Print only if "free" is present:

```
txt = "The best things in life are free!"
if "free" in txt:
  print("Yes, 'free' is present.")
```

Use it in an `if` statement:

Example

Print only if "free" is present:

```
txt = "The best things in life are free!"
if "free" in txt:
  print("Yes, 'free' is present.")
```

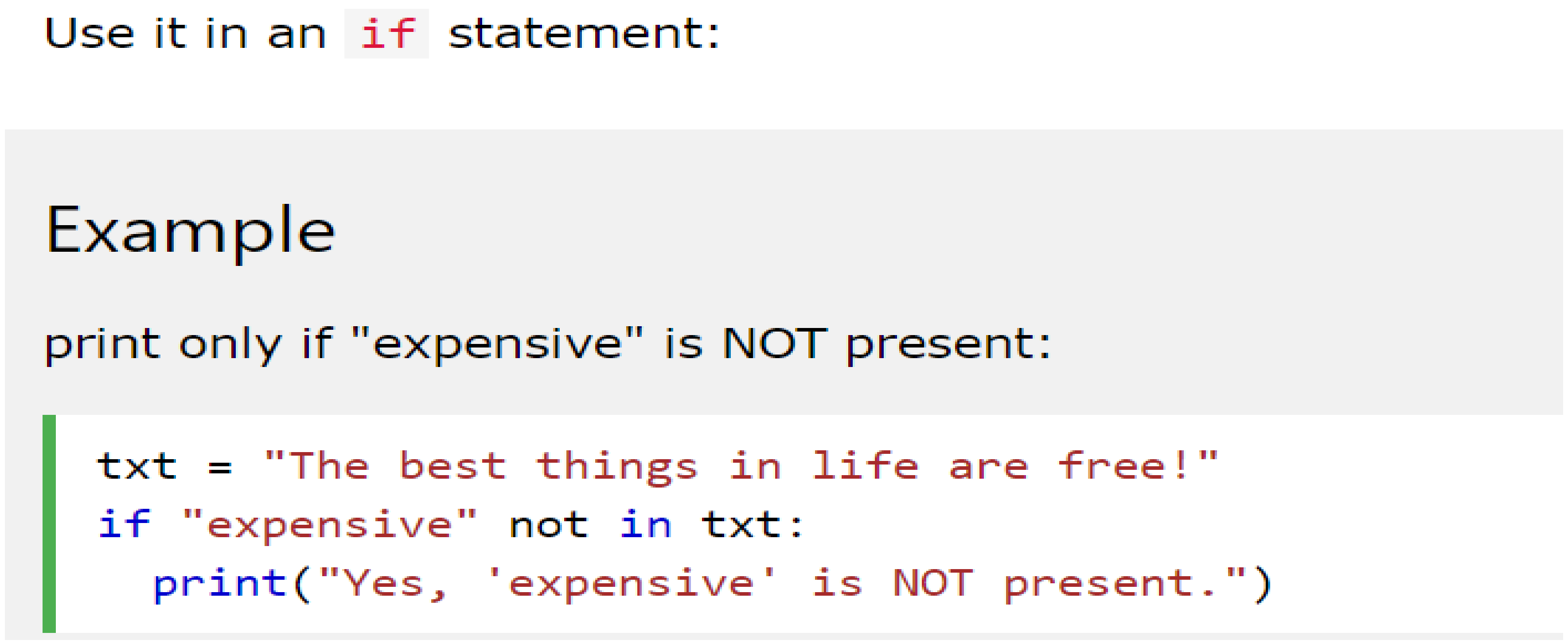
Check if NOT

To check if a certain phrase or character is NOT present in a string, we can use the keyword `not in`.

Example

Check if "expensive" is NOT present in the following text:

```python
txt = "The best things in life are free!"
print("expensive" not in txt)
```



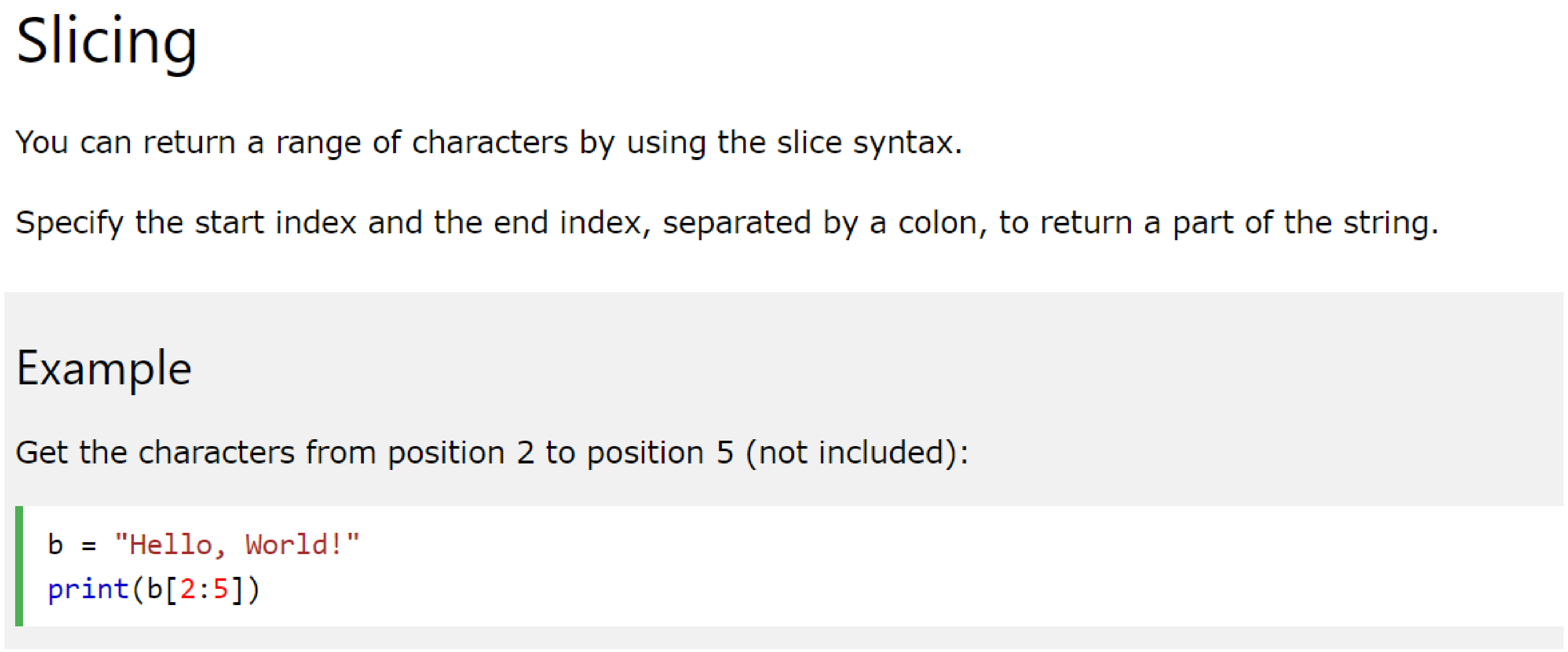
Use it in an `if` statement:

Example

print only if "expensive" is NOT present:

```
txt = "The best things in life are free!"
if "expensive" not in txt:
  print("Yes, 'expensive' is NOT present.")
```

Python - Slicing Strings



Slicing

You can return a range of characters by using the slice syntax.

Specify the start index and the end index, separated by a colon, to return a part of the string.

Example

Get the characters from position 2 to position 5 (not included):

```
b = "Hello, World!"
print(b[2:5])
```

Note: The first character has index 0.

Slice From the Start

By leaving out the start index, the range will start at the first character:

Example

Get the characters from the start to position 5 (not included):

```
b = "Hello, World!"
print(b[:5])
```

Slice To the End

By leaving out the *end* index, the range will go to the end:

Example

Get the characters from position 2, and all the way to the end:

```
b = "Hello, World!"
print(b[2:])
```

Negative Indexing

Use negative indexes to start the slice from the end of the string:

Example

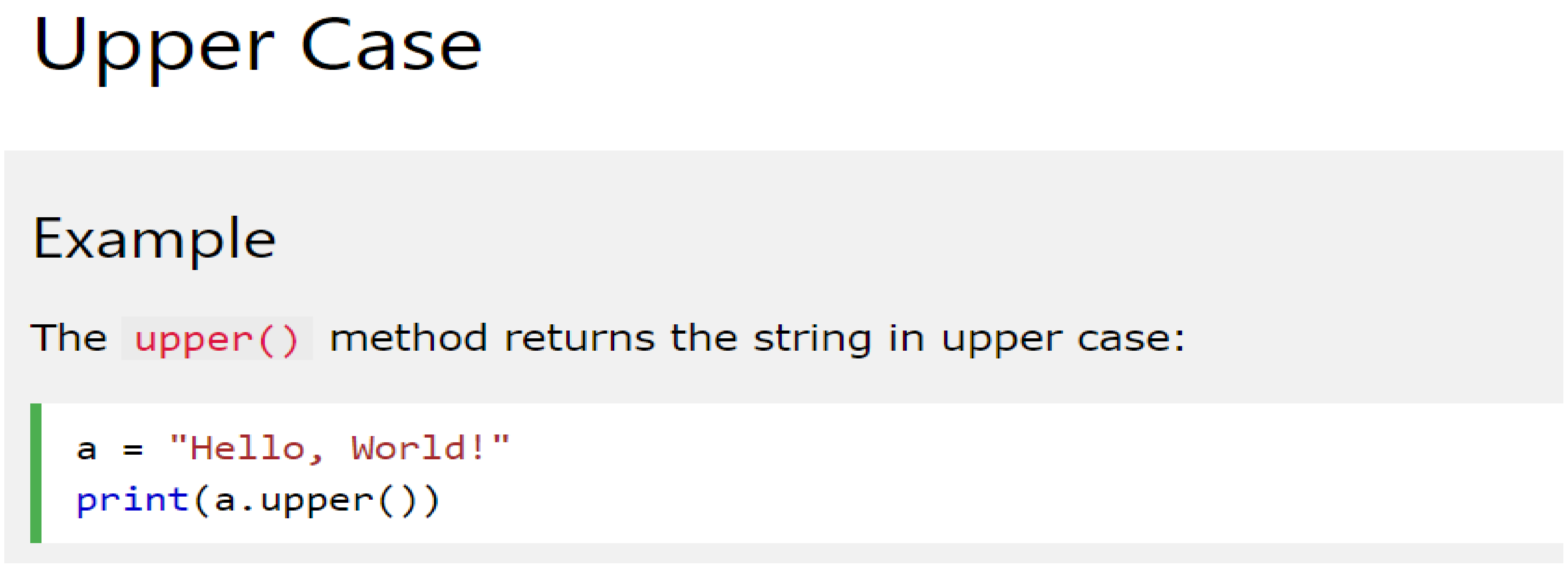
Get the characters:

From: "o" in "World!" (position -5)

To, but not included: "d" in "World!" (position -2):

```python
b = "Hello, World!"
print(b[-5:-2])
```

Python - Modify Strings



Upper Case

Example

The `upper()` method returns the string in upper case:

```python
a = "Hello, World!"
print(a.upper())
```

Lower Case

Example

The `lower()` method returns the string in lower case:

```
a = "Hello, World!"
print(a.lower())
```

Remove Whitespace

Whitespace is the space before and/or after the actual text, and very often you want to remove this space.

Example

The `strip()` method removes any whitespace from the beginning or the end:

```
a = " Hello, World! "
print(a.strip()) # returns "Hello, World!"
```

Replace String

Example

The `replace()` method replaces a string with another string:

```
a = "Hello, World!"
print(a.replace("H", "J"))
```

Split String

The `split()` method returns a list where the text between the specified separator becomes the list items.

Example

The `split()` method splits the string into substrings if it finds instances of the separator:

```
a = "Hello, World!"
print(a.split(",")) # returns ['Hello', ' World!']
```

Exercise:

Use the `len` method to print the length of the string.

```
x = "Hello World"
print(        )
```

Exercise:

Get the first character of the string `txt`.

```
txt = "Hello World"
x = 
```

Exercise:

Get the characters from index 2 to index 4 (`llo`).

```
txt = "Hello World"
x = 
```

Exercise:

Return the string without any whitespace at the beginning or the end.

```
txt = " Hello World "
x = 
```

Exercise:

Convert the value of `txt` to upper case.

```
txt = "Hello World"
txt = 
```

Exercise:

Replace the character `H` with a `J`.

```
txt = "Hello World"
txt = txt.          (    ,     )
```

THANKS