

# Longest Substring without Repeating Characters

**Question:** Given a string, find the length of the longest substring without repeating characters. For example:

Given "abcabcbb", the answer is "abc", which the length is 3.

Given "bbbbbb", the answer is "b", with the length of 1.

Given "pwwkew", the answer is "wke", with the length of 3.

## Solutions:

```
class Solution:
```

```
    # @return an integer
```

```
    def lengthOfLongestSubstring(self, s):
```

```
        lastRepeating = -1
```

```
        longestSubstring = 0
```

```
        positions = {}
```

```
        for i in range(0, len(s)):
```

```
            if s[i] in positions and lastRepeating < positions[s[i]]:
```

```
                lastRepeating = positions[s[i]]
```

```
            if i - lastRepeating > longestSubstring:
```

```
                longestSubstring = i - lastRepeating
```

```
            positions[s[i]] = i
```

```
        return longestSubstring
```

```
Solution().lengthOfLongestSubstring("abcabcbb")
```