

Cómo usar la función `re.search()` en Python

Example 1: Basic Usage of `re.search()`

```
import re

# Example 1: Basic usage of re.search()
pattern = r"\bhello\b"
text = "Say hello to the world!"

match = re.search(pattern, text)

if match:
    print("Found:", match.group())
else:
    print("Not found.")
```

Example 2: Using `re.search()` with groups

```
import re

# Example 2: Using re.search() with groups
pattern = r"(\d{4})-(\d{2})-(\d{2})"
text = "The date is 2023-08-10."

match = re.search(pattern, text)

if match:
    year, month, day = match.groups()
```

```
print(f"Year: {year}, Month: {month}, Day: {day}")
```

Example 3: re.search() with Case insensitivity

```
import re

# Example 3: re.search() with case insensitivity
pattern = r"python"
text = "I love Python programming!"

match = re.search(pattern, text, re.IGNORECASE)

if match:
    print("Found:", match.group())
else:
    print("Not found.")
```

Example 4: re.search() with a complex pattern

```
import re

# Example 4: re.search() with a complex pattern
pattern = r"\b\d{3}-\d{2}-\d{4}\b"
text = "My SSN is 123-45-6789."

match = re.search(pattern, text)

if match:
    print("Found SSN:", match.group())
else:
    print("SSN not found.")
```

Example 5: Searching for multiple patterns

```
import re

# Example 5: Searching for multiple patterns
pattern = r"cat|dog"
text = "I have a cat and a dog."

match = re.search(pattern, text)

if match:
    print("Found:", match.group())
else:
    print("Not found.")
```

Example 6: Matching the beginning of a string

```
import re

# Example 6: Matching the beginning of a string
pattern = r"^Hello"
text = "Hello, how are you?"

match = re.search(pattern, text)

if match:
    print("The string starts with 'Hello'.")
else:
    print("The string does not start with 'Hello'.")
```

Example 7: Matching the end of a

string

```
import re

# Example 7: Matching the end of a string
pattern = r"world!$"
text = "Hello, world!"

match = re.search(pattern, text)

if match:
    print("The string ends with 'world!'")
else:
    print("The string does not end with 'world!'")
```

Example 8: Ignoring whitespaces in a pattern

```
import re

# Example 8: Ignoring whitespaces in a pattern
pattern = r"\d{3} \s*-\s* \d{2} \s*-\s* \d{4}"
text = "123 - 45 - 6789"

match = re.search(pattern, text, re.VERBOSE)

if match:
    print("Found SSN with flexible spaces:", match.group())
else:
    print("SSN not found.")
```

Example 9: Finding overlapping matches

```
import re

# Example 9: Finding overlapping matches
pattern = r"(?=(\d{2}))"
text = "12345"

matches = re.finditer(pattern, text)

for match in matches:
    print("Found overlapping match:", match.group(1))
```

Example 10: Using re.search() with a dictionary

```
import re

# Example 10: Using re.search() with a dictionary
pattern_dict = {
    "email": r"\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b",
    "phone": r"\b\d{3}[-.]?\d{3}[-.]?\d{4}\b"
}

text = "Contact me at john.doe@example.com or call 123-456-7890."

for key, pattern in pattern_dict.items():
    match = re.search(pattern, text)
    if match:
        print(f"Found {key}: {match.group()}")
```

✓
0 s



```
import re

# Example 10: Using `re.search()` with a dictionary
pattern_dict = {
    "email": r"\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b",
    "phone": r"\b\d{3}[-.]?\d{3}[-.]?\d{4}\b"
}

text = "Contact me at john.doe@example.com or call 123-456-7890."

for key, pattern in pattern_dict.items():
    match = re.search(pattern, text)
    if match:
        print(f"Found {key}: {match.group()}")
```



Found email: john.doe@example.com
Found phone: 123-456-7890