Work it out in Terminal:

>>> name="helloworld"

>>> name[0:5]

'hello'

>>> name[0:6]

'hellow'

>>> name[0:-5]

'hello'

>>> name[:-5]

'hello'

>>> name[:-3]

'hellowo'

>>> name[:-10]

‘h’

>>> name[:-9]

'he'

>>> name[:-1]

'helloworl'

>>> name[:0]

>>> name[:-1]

'helloworl'

>>> name[1:5:2]

'el'

>>> name[0:5:2]

'hlo'

>>> name[0::2]

'hlool'

>>> name[0:9]

'helloworl'

>>> name[0:10]

'helloworld'

>>> n='vit'

>>> b='university'

>>> c=n+b

'vituniversity'

>>> n=123

>>> n 123

>>> n='vituniversity'

>>> len(n)

13

Python program that uses for-loop on strings

```
s = "abc"
# Loop over string.
for c in s:
    print(c)
# Loop over string indexes.
for i in range(0, len(s)):
    print(s[i])
```

Python program that uses in-operator

```
s = "dot net perls"

# Use the in-operator.

if "dot" in s:

    print("dot")

if "perls" in s:

    print("perls")

if " " in s:

    print("space")
```

Python program that adds, multiplies strings

```
s = "abc?"

# Add two strings together.

add = s + s

print(add)

# Multiply a string.

product = s * 3
```

```
print(product)
```

Python program that uses count

```
value = "finnegans wake"

# Count this substring.

print(value.count("n"))

# Count substring in indexes 0 to 6.

print(value.count("n", 0, 6))
```

Python that uses startswith, endswith

```
# Input string.

s = "voorheesville"

if s.startswith("voo"):
    print("1")

if s.endswith("ville"):
    print("2")

if s.startswith("stuy"):
    # Not reached.
```

```
    print("3")
```

Python that uses ljust, rjust

```
s = "Paris"

# Justify to left, add periods.

print(s.ljust(10))

# Justify to right.

print(s.rjust(10))

print(s.rjust(10, "."))

print(s.ljust(10, "."))
```

Python that uses replace

```
value = "aabc"

# Replace a substring with another.

result = value.replace("bc", "yz")

print(result)

# Replace the first occurrence with a substring.

result = value.replace("a", "x", 1)
```

```
print(result)
```

Python that tests string equality

```
value = "CAT"

if value == "cat":
    print("A") # Not reached.

if value == "CAT":
    print("B")

if str.casefold(value) == "cat":
    print("C")™™™™•

if str.lower(value) == "cat":
    print("D")
```

Python that uses raw string

```
# In a raw string "\" characters do not escape.

raw = r"\directory\123"

val = "\directory\123"

print(raw)
```

```
print(val)
```

Python that uses ascii built-in

```
# This string contains an umlaut.
value = "Düsseldorf"
print(value)
# Display letter with escaped umlaut.
print(ascii(value))
```

Python that uses string.digits

```
import string
# Loop over digits using string.digits constant.
for digit in string.digits:
    print(digit)
```

Python that uses string. Punctuation, whitespace

```
import string
# Display punctuation.
print(string.punctuation)•
```

The space is included in string.whitespace.

print(" " in string.whitespace)

Replacing strings

#!/usr/bin/python

a = "I saw a wolf in the forest. A lonely wolf."

b = a.replace("wolf", "fox")

print(b)

c = a.replace("wolf", "fox", 1)

print(c)Splitting, joining stringsnums = "1,5,6,8,2,3,1,9"

k = nums.split(",")

print(k)

l = nums.split(",", 5) //5 is the maximum splits allowed

print(l)

m = nums.rsplit(",", 3)

print(m)

nums = "1,5,6,8,2,3,1,9"

```
n = nums.split(",")

print(n)

m = ':'.join(n)

print(m)Working with casea = "ZetCode"

print(a.upper())

print(a.lower())

print(a.swapcase())

print(a.title() )Operations on stringssentence = "There are 22 apples"

alphas = 0

digits = 0

spaces = 0

for i in sentence:

   if i.isalpha():

      alphas += 1

   if i.isdigit():

      digits += 1
```

```
    if i.isspace():

        spaces += 1

print("There are", len(sentence), "characters")

print("There are", alphas, "alphabetic characters")

print("There are", digits, "digits")

print("There are", spaces, "spaces")
```

Escape SequenceDescription\newlineBackslash and newline ignored\\Backslash\'Single quote\"Double quote\aASCII Bell\bASCII Backspace\fASCII Formfeed\nASCII Linefeed\rASCII Carriage Return\tASCII Horizontal Tab\vASCII Vertical Tab\oooCharacter with octal value ooo\xHHCharacter with hexadecimal value HH