Strings

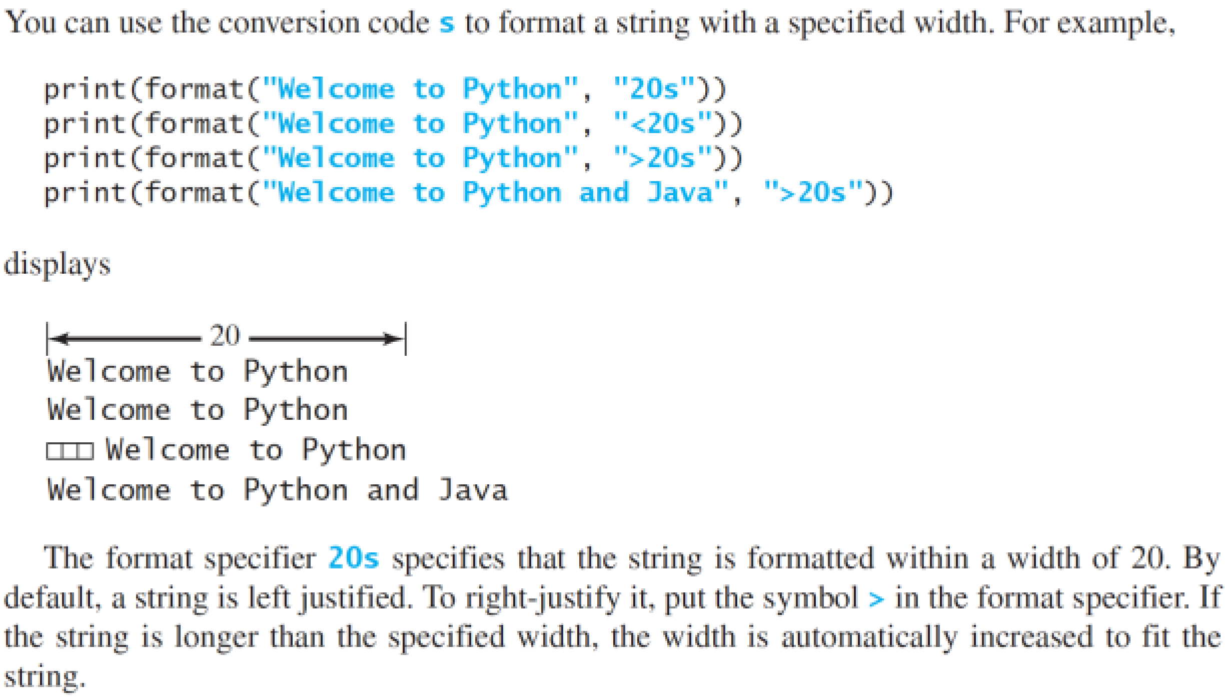
name = “james”
print(name.upper() # will print JAMES
print(len(name)) # will print 5

name2 = “james “
print(name.strip()) #remove blanks and print james

name = “college of engineering”
print(name[0:7]) # will print college
print(name.count(“e”)) # will print 5

color = “red,blue,green”
r,b,g = x.split(“,”)
print (r) # will print red
print (b) # blue
print (g) # green

Formatting Strings



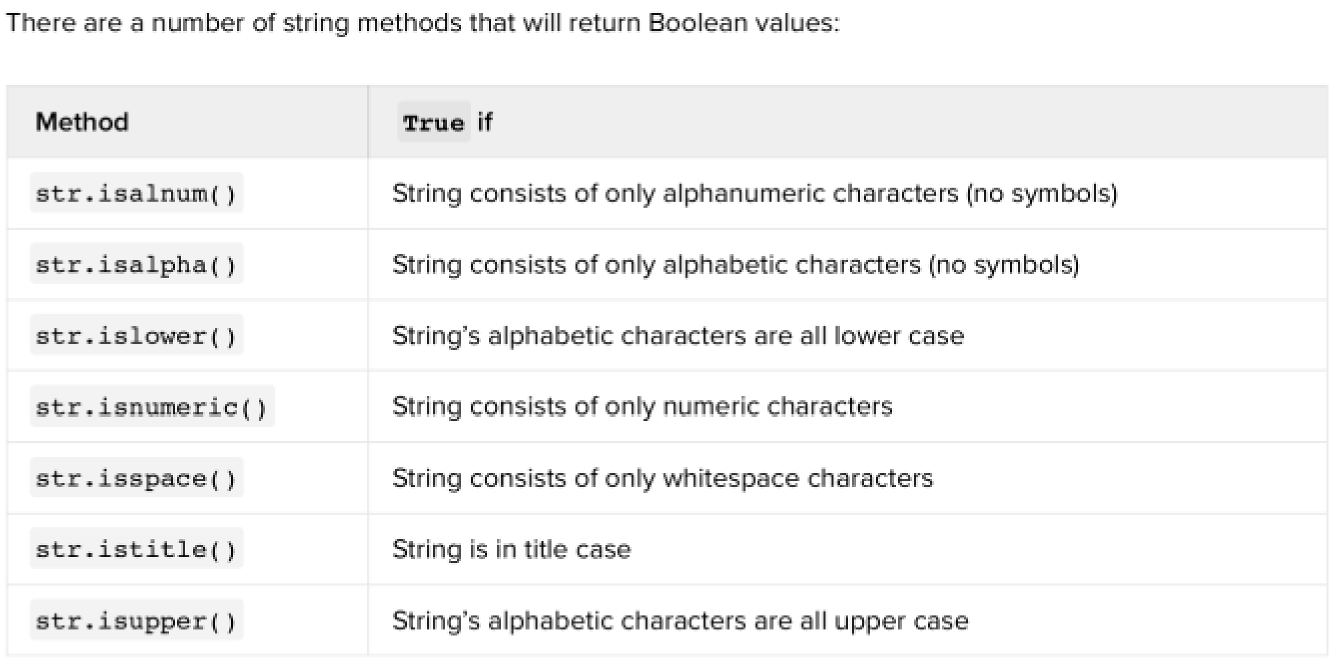
You can use the conversion code **s** to format a string with a specified width. For example,

```
print(format("Welcome to Python", "20s"))
print(format("Welcome to Python", "<20s"))
print(format("Welcome to Python", ">20s"))
print(format("Welcome to Python and Java", ">20s"))
```

displays

```
|←——— 20 ———→|
Welcome to Python
Welcome to Python
□□□Welcome to Python
Welcome to Python and Java
```

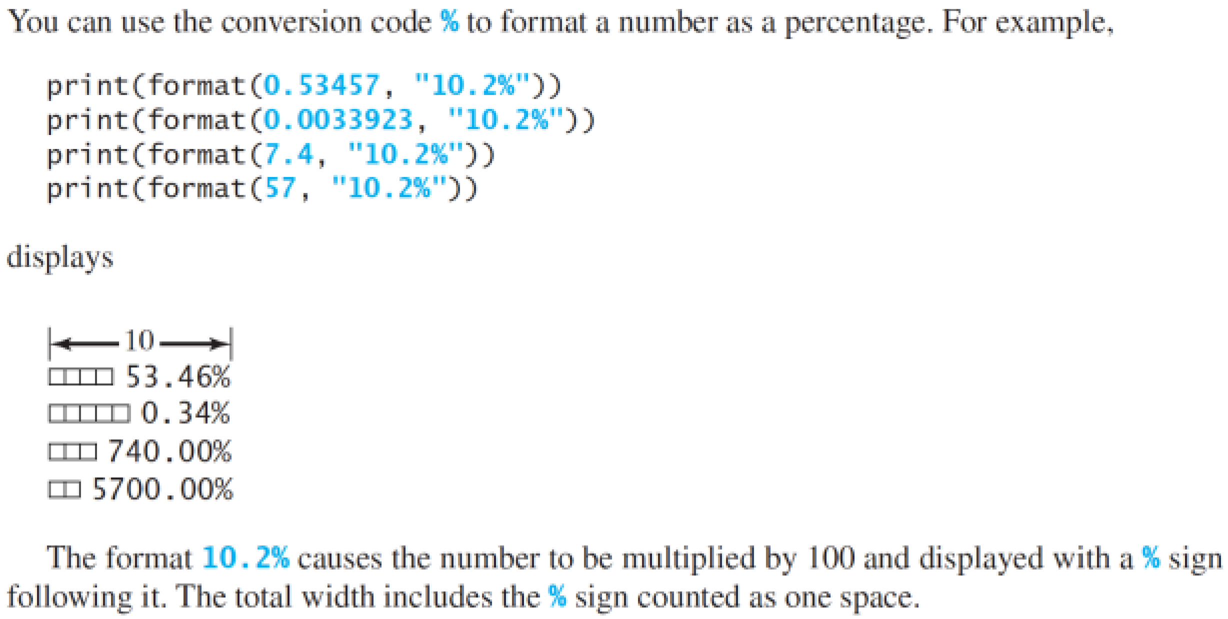
The format specifier **20s** specifies that the string is formatted within a width of 20. By default, a string is left justified. To right-justify it, put the symbol **>** in the format specifier. If the string is longer than the specified width, the width is automatically increased to fit the string.



There are a number of string methods that will return Boolean values:

Method	`True` if
`str.isalnum()`	String consists of only alphanumeric characters (no symbols)
`str.isalpha()`	String consists of only alphabetic characters (no symbols)
`str.islower()`	String's alphabetic characters are all lower case
`str.isnumeric()`	String consists of only numeric characters
`str.isspace()`	String consists of only whitespace characters
`str.istitle()`	String is in title case
`str.isupper()`	String's alphabetic characters are all upper case

Formatting as a Percentage



You can use the conversion code `%` to format a number as a percentage. For example,

```
print(format(0.53457, "10.2%"))
print(format(0.0033923, "10.2%"))
print(format(7.4, "10.2%"))
print(format(57, "10.2%"))
```

displays

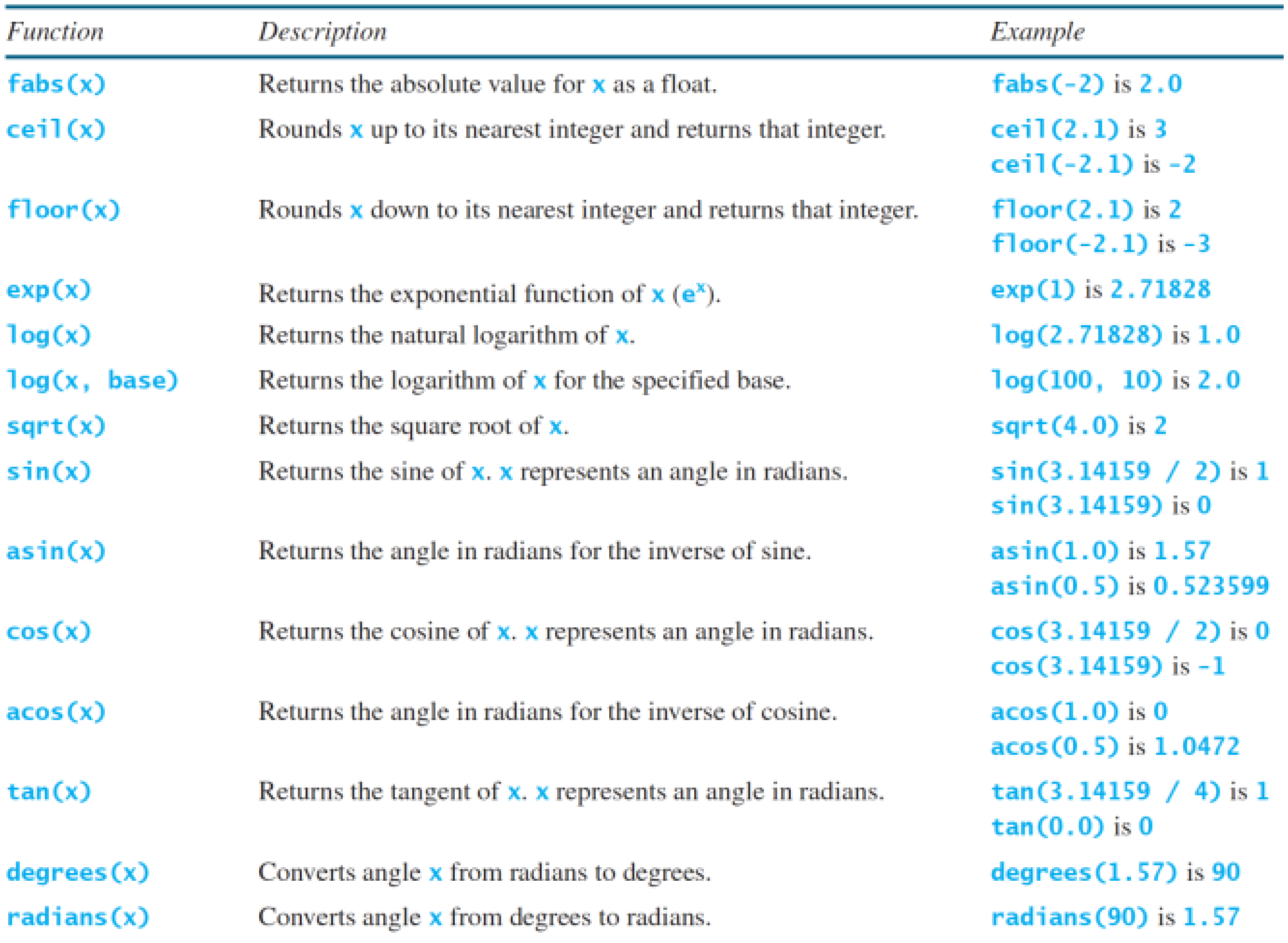
```
|←—10—→|
□□□□53.46%
□□□□□0.34%
□□□740.00%
□□5700.00%
```

The format `10.2%` causes the number to be multiplied by 100 and displayed with a `%` sign following it. The total width includes the `%` sign counted as one space.

Math

```
from math import *

i = 17.34989436516001
print (round(i,4))    # 17.3499
print (sqrt(4.0))      # 2.0
pow (2, 3)            # 8
```



Function	*Description*	*Example*
fabs(x)	Returns the absolute value for x as a float.	fabs(-2) is 2.0
ceil(x)	Rounds x up to its nearest integer and returns that integer.	ceil(2.1) is 3 ceil(-2.1) is -2
floor(x)	Rounds x down to its nearest integer and returns that integer.	floor(2.1) is 2 floor(-2.1) is -3
exp(x)	Returns the exponential function of x (e^x).	exp(1) is 2.71828
log(x)	Returns the natural logarithm of x.	log(2.71828) is 1.0
log(x, base)	Returns the logarithm of x for the specified base.	log(100, 10) is 2.0
sqrt(x)	Returns the square root of x.	sqrt(4.0) is 2
sin(x)	Returns the sine of x. x represents an angle in radians.	sin(3.14159 / 2) is 1 sin(3.14159) is 0
asin(x)	Returns the angle in radians for the inverse of sine.	asin(1.0) is 1.57 asin(0.5) is 0.523599
cos(x)	Returns the cosine of x. x represents an angle in radians.	cos(3.14159 / 2) is 0 cos(3.14159) is -1
acos(x)	Returns the angle in radians for the inverse of cosine.	acos(1.0) is 0 acos(0.5) is 1.0472
tan(x)	Returns the tangent of x. x represents an angle in radians.	tan(3.14159 / 4) is 1 tan(0.0) is 0
degrees(x)	Converts angle x from radians to degrees.	degrees(1.57) is 90
radians(x)	Converts angle x from degrees to radians.	radians(90) is 1.57