Built-in Functions

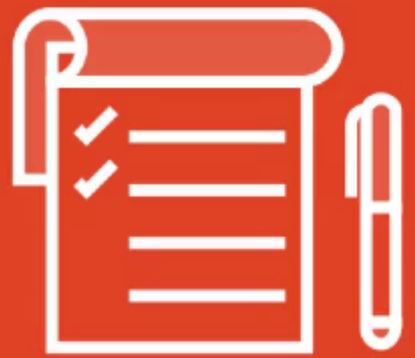
Austin Bingham
COFOUNDER - SIXTY NORTH

@austin_bingham

Robert Smallshire
COFOUNDER - SIXTY NORTH

@robsmallshire

Overview



Rounding and absolute values

Literal forms for integers using different bases

Create string versions of integers in different bases

Built-in Functions for Numbers

abs()

Returns the magnitude of a number

For non-complex numbers, this is simply the number's value without regard to its sign

abs() and round()

```
5.0
>>> abs(Decimal(-5))
Decimal('5')
>>> abs(Fraction(-5, 1))
Fraction(5, 1)
>>> abs(complex(0, -5))
5.0
>>> round(0.2812, 3)
0.281
>>> round(0.625, 1)
0.6
>>> round(1.5)
2
>>> round(2.5)
2
>>> round(867)
867
>>> round(53.09)
53
>>> round(Decimal('3.25'), 1)
Decimal('3.2')
>>> round(Fraction(57, 100), 2)
Fraction(57, 100)
>>>
```

Rounding Floats

`float` uses a binary representation

`round()` is fundamentally a decimal operation

Rounding 2.675 to two places should yield 2.68

Rounding Floats

```
>>> round(2.675, 2)
2.67
>>> from decimal import Decimal
>>> round(Decimal('2.675'), 2)
Decimal('2.68')
>>>
```

Base Conversions

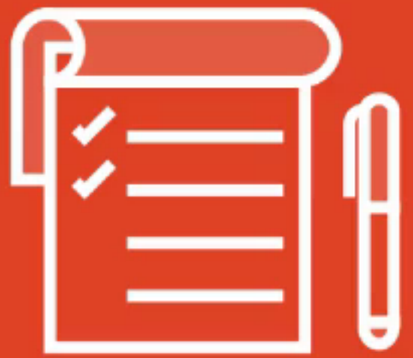
```
>>> 0b101010
42
>>> 0o52
42
>>> 0x2a
42
>>> bin(42)
'0b101010'
>>> oct(42)
'0o52'
>>> hex(42)
'0x2a'
>>> hex(42)[2:]
'2a'
>>> int("2a", base=16)
42
>>>
```

The valid values for the base argument are zero and 2-36 inclusive.

Base Conversions

```
>>> int("acghd", base=18)
1125247
>>> int("0b111000", base=2)
56
>>> int("0o664", base=0)
436
>>>
```

Summary



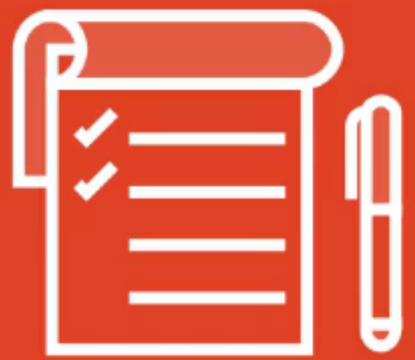
`abs()` calculates the magnitude of a number

`round()` rounds to a given number of decimal digits

`round()` does not work for `complex`

It can give surprising results for `float`

Summary



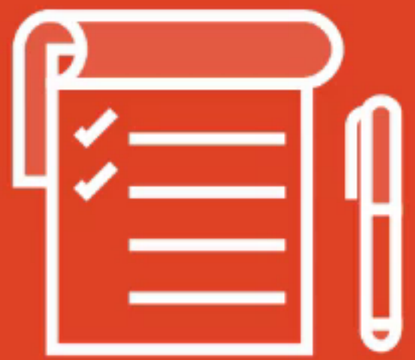
Python has literal forms for binary, octal, and hexadecimal

`bin()` produces the binary representation of an integer

`oct()` produces the octal representation of an integer

`hex()` produces the hexadecimal representation of an integer

Summary



`int` accepts an optional `base` argument

`base` specifies the base to use when interpreting the string

`base` can be any value from 2-36

If `base` is zero, Python uses the string's prefix or decimal