

String Representation of Objects



Robert Smallshire
COFOUNDER - SIXTY NORTH
[@robsmallshire](https://twitter.com/robsmallshire)



Austin Bingham
COFOUNDER - SIXTY NORTH
[@austin_bingham](https://twitter.com/austin_bingham)

Overview



Three built-in functions:

- r = repr(obj)
- s = str(obj)
- f = format(obj)

Customization gives:

- maintainability
- debuggability
- usability

position > position.py

position.py ×

```
1 class Position:
2
3     def __init__(self, latitude, longitude):
4         if not (-90 <= latitude <= +90):
5             raise ValueError(f"Latitude {latitude} out of range")
6
7         if not (-180 <= longitude <= +180):
8             raise ValueError(f"Longitude {longitude} out of range")
9
10    self._latitude = latitude
```

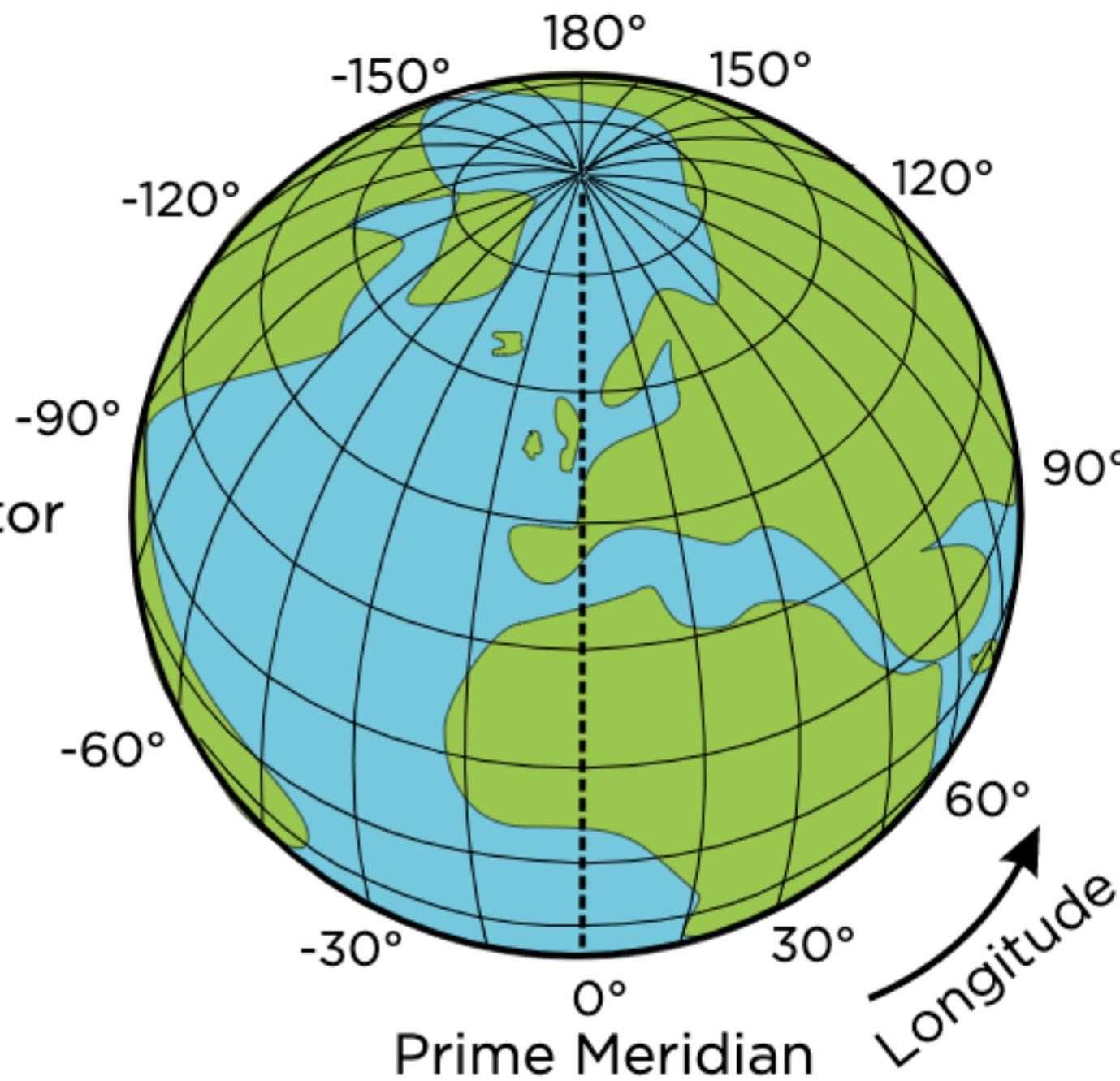
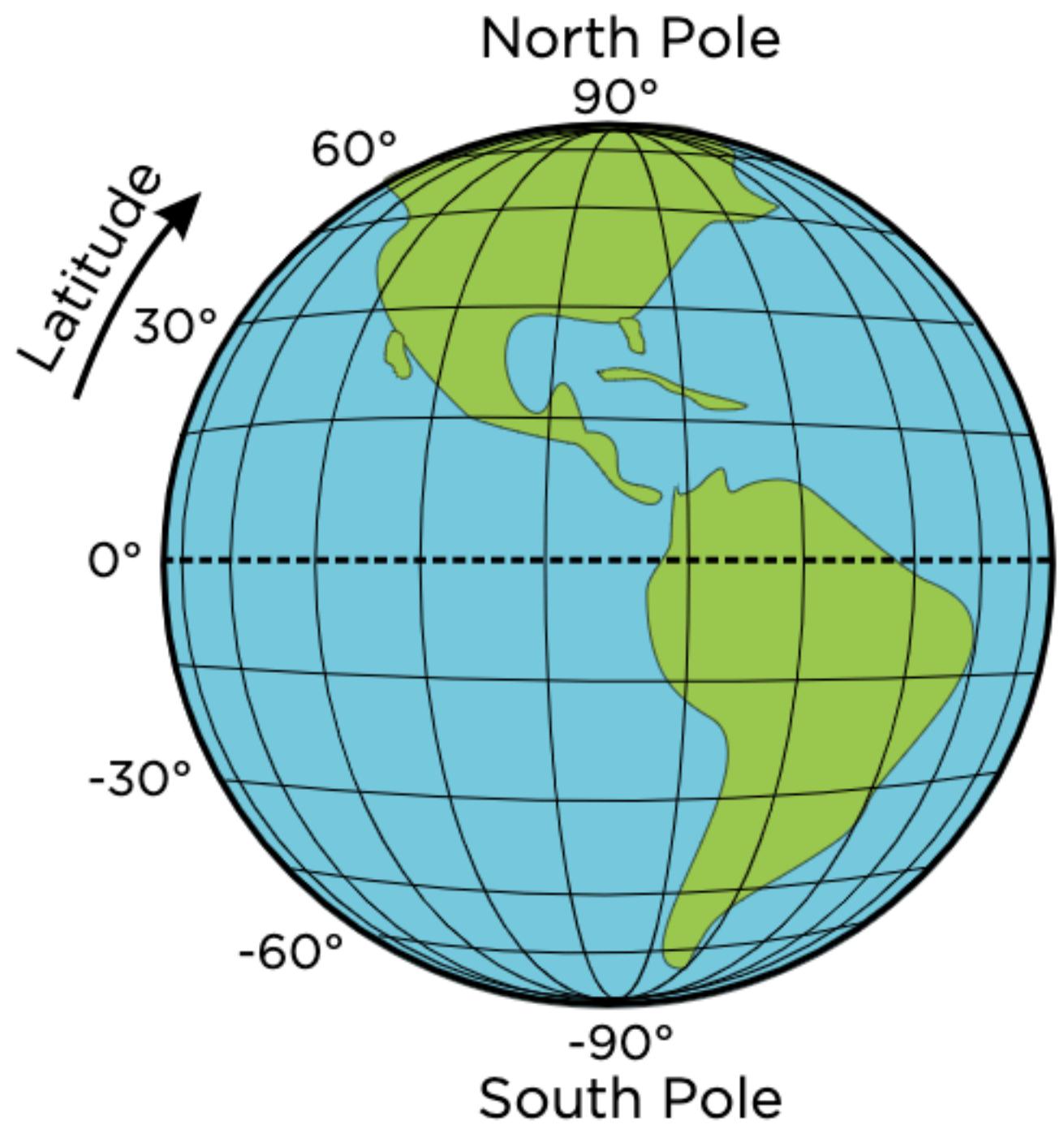
Position

Python Console ×

```
<position.Position object at 0x104a9efa0>
>>> str(oslo)
<position.Position object at 0x104a9efa0>
>>> format(oslo)
<position.Position object at 0x104a9efa0>
>>> dir(object)
['__class__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__gt__',
 '__hash__', '__init__', '__init_subclass__', '__le__', '__lt__', '__ne__', '__new__', '__reduce__', '__reduce_ex__',
 '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__']

>>>
```

Latitude and Longitude



Customizing repr()

position > position.py

position.py

```
21     def __repr__(self):
22         return f"{typename(self)}(latitude={self.latitude}, longitude={self.longitude})"
23
24
25 class EarthPosition(Position):
26     pass
27
28
29 class MarsPosition(Position):
30     pass
```

Python Console

```
>>> mauna_keia
EarthPosition(latitude=19.82, longitude=-155.47)
>>> olympus_mons = MarsPosition(18.65, -133.8)
>>> olympus_mons
MarsPosition(latitude=18.65, longitude=-133.8)
>>> str(olympus_mons)
'MarsPosition(latitude=18.65, longitude=-133.8)'
>>> format(olympus_mons)
'MarsPosition(latitude=18.65, longitude=-133.8'

>>>
```

Consider the Target Audience



For whom is `repr()` intended?



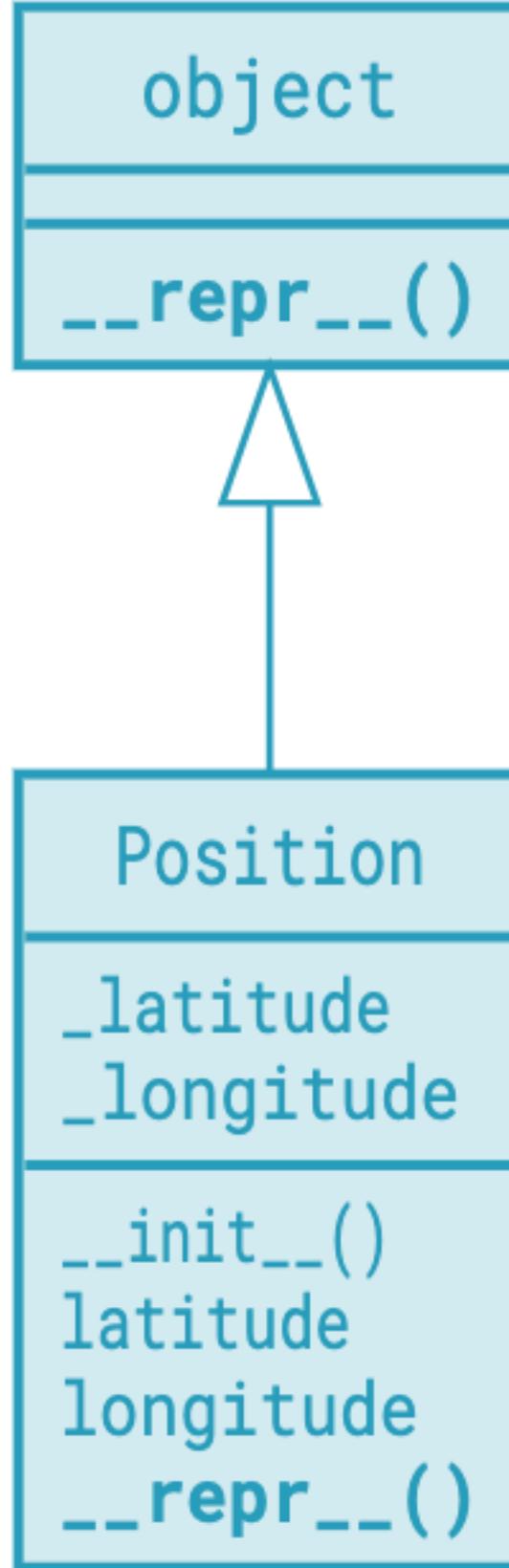
Conventions for Good `_repr_` Results



Include necessary state, but be prepared to compromise.



Format as constructor invocation source code.



Overriding `__repr__()`

The default `__repr__` inherited from `object` is not much use.

Override `__repr__` to return a more useful string, which ideally formatted as source code for a constructor call.

You should almost always
override `__repr__()` in your
classes.

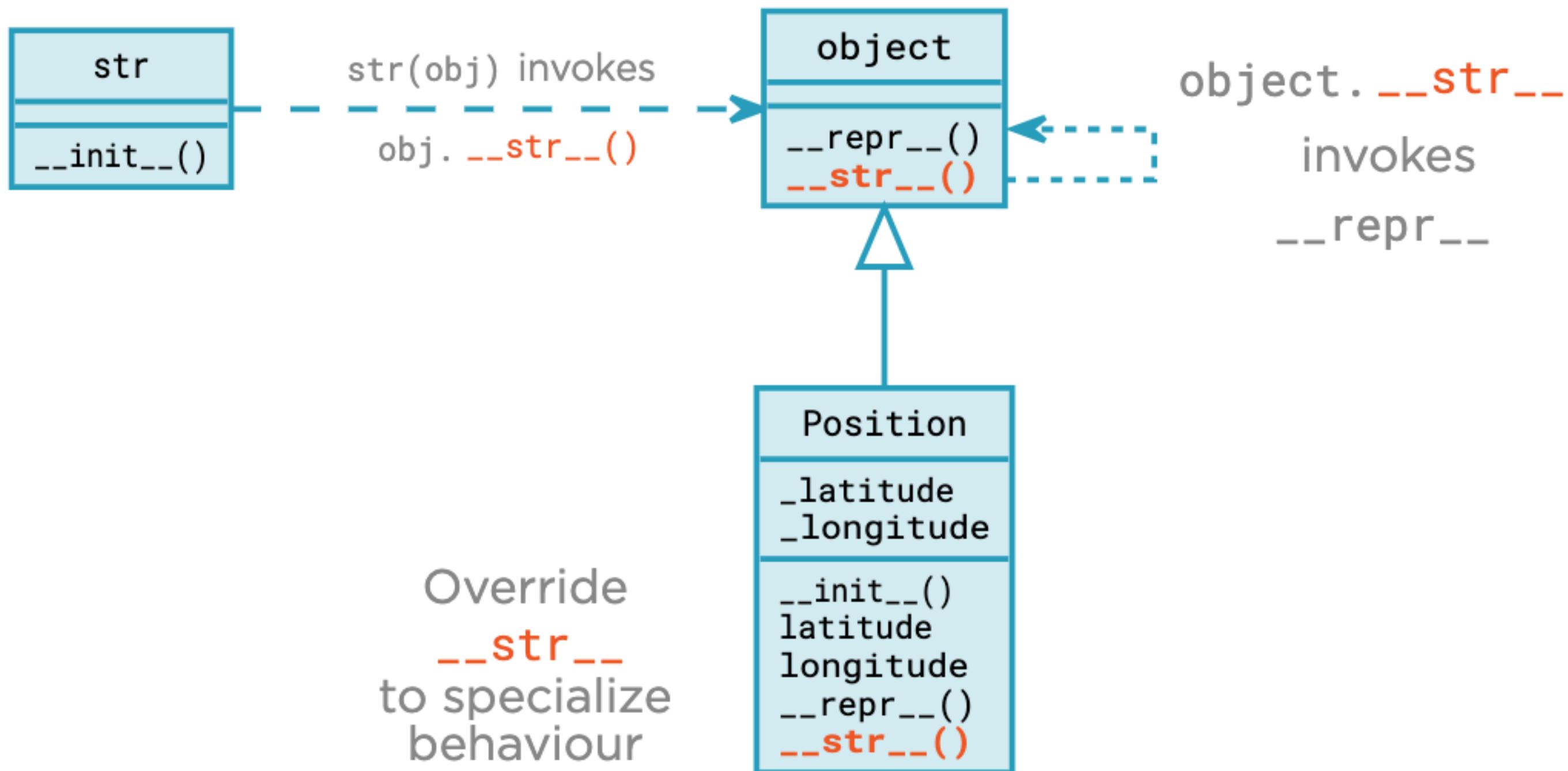
Customizing str()

What Is the str() Function?

`str(obj)`

The string class is callable

String Constructor Delegation



Consider the Target Audience



For whom is `str()` intended?

`str()` Is for System Consumers



**Users, people.
In user interfaces.**



Other systems.

Think of the User

~~Position~~~~latitude=-34.8, longitude=14.4~~

Geographic Positions

77.5° S, 167.2° E



Mount Erebus, Antarctica

position > position.py

position.py ×

```
27     return "E" if self.longitude >= 0 else "W"
28
29 def __repr__(self):
30     return f"{typename(self)}(latitude={self.latitude}, longitude={self.longitude})"
31
32 def __str__(self):
33     return (
34         f"{abs(self.latitude)}° {self.latitude_hemisphere}, "
35         f"{abs(self.longitude)}° {self.longitude_hemisphere}"
36     )
```

Position > __str__()

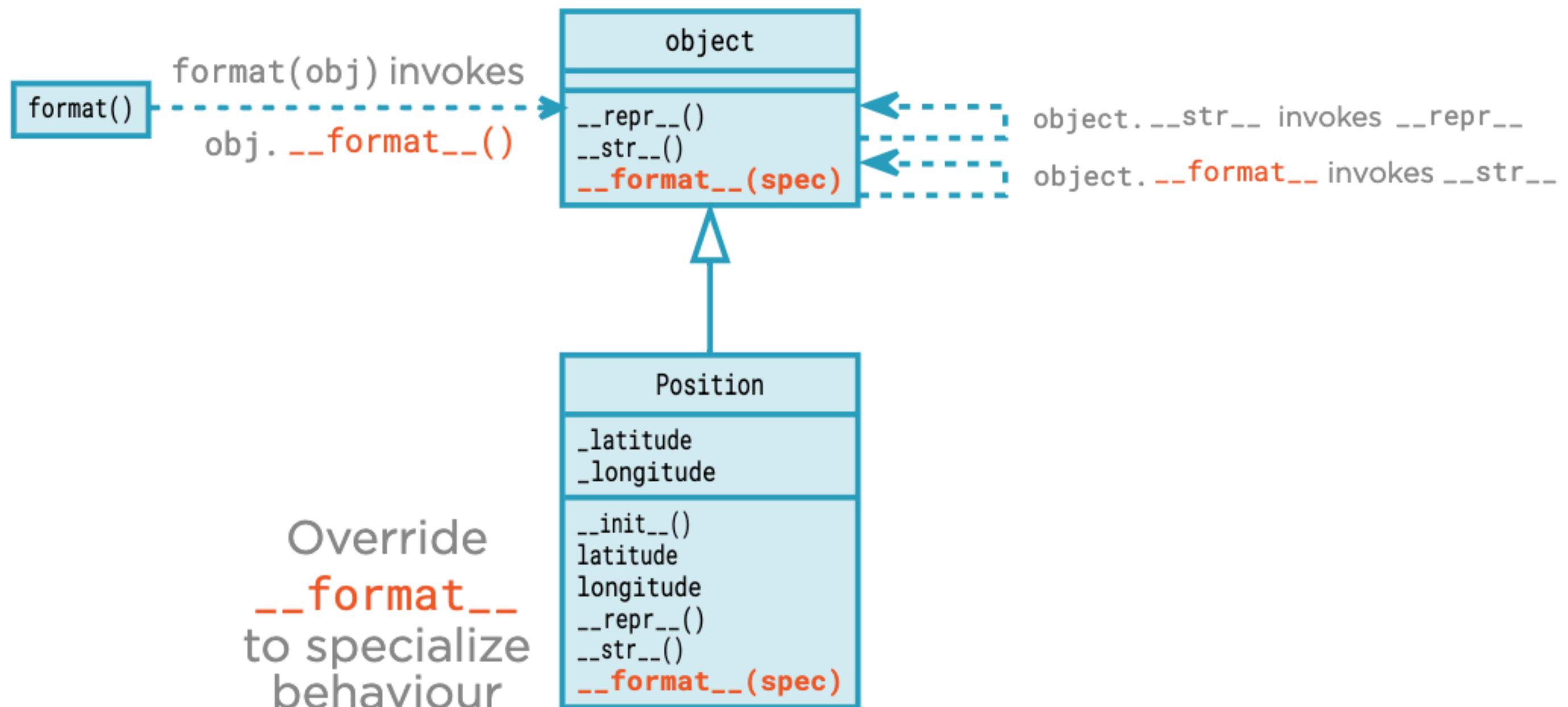
Python Console ×

```
'77.5° S, 167.2° E'
>>> print("Mount Erebus is located at", mount_erebus)
Mount Erebus is located at 77.5° S, 167.2° E
>>> repr(mount_erebus)
'EarthPosition(latitude=-77.5, longitude=167.2)'
>>> str(mount_erebus)
'77.5° S, 167.2° E'
>>> format(mount_erebus)
'77.5° S, 167.2° E'

>>>
```

Customizing format()

format() Function Delegation



```
position.py x
33
34     return (
35         f"{abs(self.latitude)}° {self.latitude_hemisphere}, "
36         f"{abs(self.longitude)}° {self.longitude_hemisphere}"
37     )
38
39     def __format__(self, format_spec):
40         return "FORMATTED POSITION"
41
42     class EarthPosition(Position):
43         pass
```

Python Console 

```
'EarthPosition(latitude=-32.7, longitude=-70.1)'  
=> str(aconcagua)  
'32.7° S, 70.1° W'  
>>> format(aconcagua)  
'FORMATTED POSITION'  
>>> f"The highest mountain in South America is located at {aconcagua}"  
'The highest mountain in South America is located at FORMATTED POSITION'  
>>> "The highest mountain in South America is located at {}".format(aconcagua)  
'The highest mountain in South America is located at FORMATTED POSITION'  
>>>
```

Floating-point Format Specifications

```
' +0.00007748091'
>>> format("Format this!", ">+20.11f")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: Unknown format code 'f' for object of type 'str'
>>> f"The conductance quantum is {q}"
'The conductance quantum is 7.748091e-05'
>>> f"The conductance quantum is {q:.6f}"
'The conductance quantum is 0.000077'
>>> f"The conductance quantum is {q:.2e}"
'The conductance quantum is 7.75e-05'
>>>
```

```
22     def __str__(self):
23         return "N" if self.latitude >= 0 else "S"
24
25     @property
26     def longitude_hemisphere(self):
27         return "E" if self.longitude >= 0 else "W"
28
29     def __repr__(self):
30         return f"{typename(self)}(latitude={self.latitude}, longitude={self.longitude})"
31
32     def __str__(self):
33         return format(self)
34
35     def __format__(self, format_spec):
36         component_format_spec = ".2f"
37         prefix, dot, suffix = format_spec.partition(".")
38         if dot:
39             num_decimal_places = int(suffix)
40             component_format_spec = f".{num_decimal_places}f"
41         latitude = format(abs(self.latitude), component_format_spec)
42         longitude = format(abs(self.longitude), component_format_spec)
43         return (
```

position > position.py

position.py X

```
35 def __format__(self, format_spec):
36     component_format_spec = ".2f"
37     prefix, dot, suffix = format_spec.partition(".")
38     if dot:
39         num_decimal_places = int(suffix)
40         component_format_spec = f".{num_decimal_places}f"
41     latitude = format(abs(self.latitude), component_format_spec)
42     longitude = format(abs(self.longitude), component_format_spec)
43     return (
44         f"{latitude}° {self.latitude_hemisphere}, "
45         f"{longitude}° {self.longitude_hemisphere}"
46     )
47
```

Position

Python Console X

```
'The everest object is EarthPosition(latitude=27.988056, longitude=86.925278)'
>>> f"The everest object is {everest!s}"
'The everest object is 27.99° N, 86.93° E'
>>> f"{everest}"
'everest=EarthPosition(latitude=27.988056, longitude=86.925278)'

>>>
```

Summary



repr() gives a string for developers
str() – the string constructor – gives a string for users
format() gives more control

Summary



`repr(obj)` delegates to `obj.__repr__()`

`str(obj)` delegates to `obj.__str__()`

**`format(obj, spec)` delegates to
`obj.__format__(spec)`**

Summary



All classes inherit default `__repr__()`,
`__str__()` and `__format__()`

The default `__repr__()` isn't very helpful

Most classes should override `__repr__()`