# Introspecting Scopes



Austin Bingham
COFOUNDER - SIXTY NORTH
@austin\_bingham



Robert Smallshire
COFOUNDER - SIXTY NORTH
@robsmallshire

#### Overview



Interact with data structures containing the global namespace

Interact with local namespaces

Leverage the fact that namespaces are implemented as standard data structures

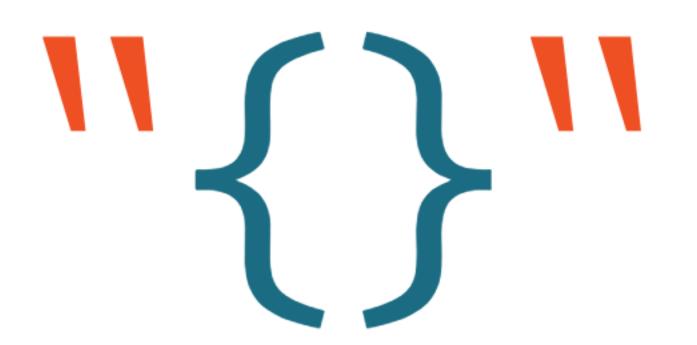
### Introspecting the Global Namespace

```
>>> globals()
{'__name__': '__main__', '__doc__': None, '__package__': None, '__loader__': <_f
rozen_importlib_external.SourceFileLoader object at 0x100920670>, '__spec__': No
ne, '__annotations__': {}, '__builtins__': <module 'builtins' (built-in)>, 'sys'
: <module 'sys' (built-in)>}
>>> a = 42
>>> globals()
{'__name__': '__main__', '__doc__': None, '__package__': None, '__loader__': <_f
rozen_importlib_external.SourceFileLoader object at 0x100920670>, '__spec__': No
ne, '__annotations__': {}, '__builtins__': <module 'builtins' (built-in)>, 'sys'
: <module 'sys' (built-in)>, 'a': 42}
>>> globals()['tau'] = 6.283185
>>> tau
6.283185
>>> tau / 2
3.1415925
>>>
```

The dictionary returned by globals() doesn't just represent the global namespace, it actually is the global namespace!

### Introspecting Local Namespaces

```
>>> locals()
{'__name__': '__main__', '__doc__': None, '__package__': None, '__loader__': <_f
rozen_importlib_external.SourceFileLoader object at 0x105b0e670>, '__spec__': No
ne, '__annotations__': {}, '__builtins__': <module 'builtins' (built-in)>, 'sys'
: <module 'sys' (built-in)>}
>>> def report_scope(arg):
from pprint import pprint as pp
x = 496
      pp(locals(), width=10)
>>> report_scope(42)
{'arg': 42,
 'pp': <function pprint at 0x105c88040>,
 'x': 496}
>>>
```



Extended call syntax allows us to unpack a dictionary into a function's keyword arguments.

str.format() accepts keyword arguments that correspond to format placeholders.

## Unpacking the Local Namespace

```
>>> name = "Joe Bloggs"
>>> age = 28
>>> country = "New Zealand"
>>> "{name} is {age} years old and is from {country}".format(**locals())
'Joe Bloggs is 28 years old and is from New Zealand'
>>>
```

## Literal String Interpolation



PEP 498 introduced a new string literal: f-strings.

f-strings interpolate names from namespaces directly into strings.

#### F-strings

```
>>> name = "Joe Bloggs"
>>> age = 28
>>> country = "New Zealand"
>>> f"{name} is {age} years old and is from {country}"
'Joe Bloggs is 28 years old and is from New Zealand'
>>>
```

#### Summary



globals() is a dictionary mapping names to objects in the global namespace

globals() actually is the global namespace

locals() returns the local namespace

We can unpack the globals() and locals() dictionaries to pass names into str.format()

It's generally better practice to use fstrings if possible