Lightning and __dunder__

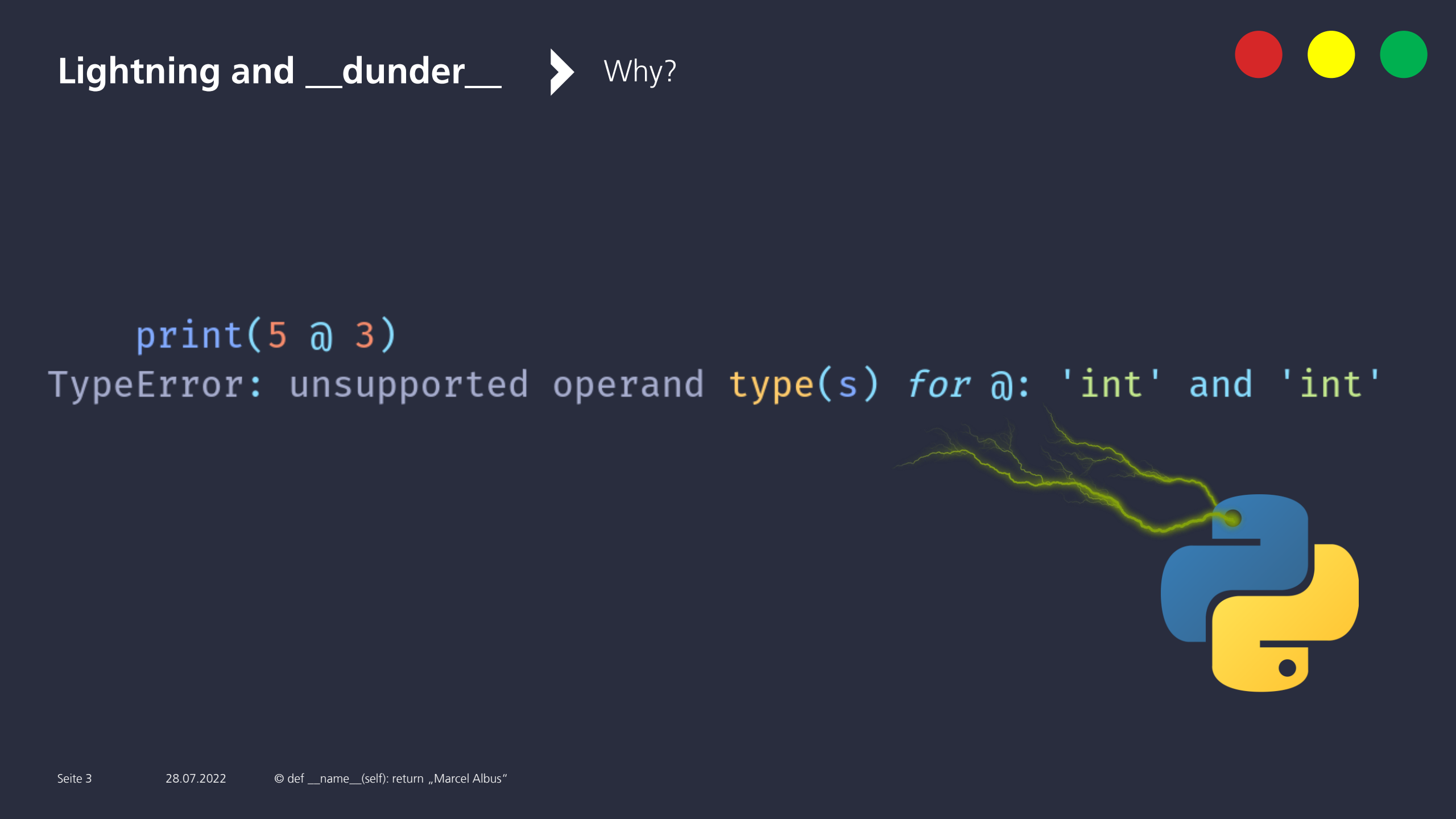
Sommerseminar 2022

Marcel Albus, Fraunhofer IPA, Stuttgart

Lightning and __dunder__ > Why?

```
print(5 + 3)
print(5 - 3)
print(5 @ 3)
```

Lightning and __dunder__ > Why?



```
    print(5 @ 3)
TypeError: unsupported operand type(s) for @: 'int' and 'int'
```

Lightning and __dunder__ > dir(int)

```
['__abs__',
 '__add__',
 '__and__',
 '__bool__',
 '__ceil__',
 '__class__',
 '__delattr__',
 '__dir__',
 '__divmod__',
 '__doc__',
 '__eq__',
 '__float__',
 '__floor__',
 '__floordiv__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__index__',
 '__init__',
 '__init_subclass__',
 '__int__',
 '__invert__',
 '__le__',
 '__lshift__',
 '__lt__',
 '__mod__',
 '__mul__',
 '__ne__',
 '__neg__',
 '__new__',
 '__or__',
 '__pos__',
 '__pow__',
 '__radd__',
 '__rand__',
 '__rdivmod__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rfloordiv__',
 '__rlshift__',
 '__rmod__',
 '__rmul__',
 '__ror__',
 '__round__',
 '__rpow__',
 '__rrshift__',
 '__rshift__',
 '__rsub__',
 '__rtruediv__',
 '__rxor__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__sub__',
 '__subclasshook__',
 '__truediv__',
 '__trunc__',
 '__xor__',
 'as_integer_ratio',
 'bit_length',
 'conjugate',
 'denominator',
 'from_bytes',
 'imag',
 'numerator',
 'real',
 'to_bytes']
```

no
“__matmul__”

Lightning and __dunder__ > Why?

Lightning and __dunder__ > What?

```python
class MyInt:
    def __init__(self, value):
        self.value = value

    def __matmul__(self, other):
        return MyInt(self.value + other.value)

    def __str__(self):
        return f"{self.value}"


print(MyInt(5) @ MyInt(3))
# >> 8
```

Lightning and __dunder__ > What?

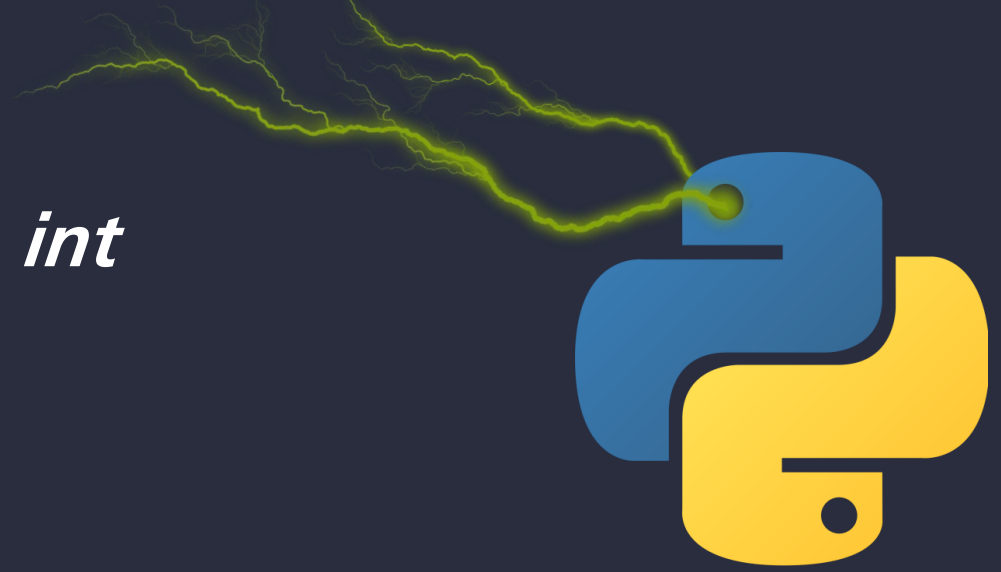
```python
class MyInt:
    def __init__(self, value):
        self.value = value

    def __matmul__(self, other):
        return MyInt(self.value + other.value)

    def __str__(self):
        return f"{self.value}"


print(MyInt(5) + MyInt(3))
# >>     print(MyInt(5) + MyInt(3))
# >> TypeError: unsupported operand type(s) for +: 'MyInt' and 'MyInt'
```

Lightning and __dunder__ > What?

```python
class Vector:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __add__(self, other):
        return Vector(self.x + other.x, self.y + other.y)

    def __str__(self):
        return f"Vector[x={self.x}, y={self.y}]"

print(Vector(2, 3) + Vector(5, 6))
# >> Vector[x = 7, y = 9]
```

Lightning and __dunder__ > __limitations__(self):

```
def __len__(self):
    return (self.x ** 2 + self.y ** 2)**0.5
```



NEEDS to return *int*

Lightning and __dunder__ > What?

```python
class Vector:
    def __init__(self, x, y):
        self.x = x
        self.y = y
        self.__n = 0

    def __add__(self, other):
        return Vector(self.x + other.x, self.y + other.y)

    def __str__(self):
        return f"Vector[x={self.x}, y={self.y}]"

    def __iter__(self):
        return self
```

```python
def __next__(self):
    self.__n += 1
    if self.__n == 1:
        return self.x
    if self.__n == 2:
        return self.y
    if self.__n == 3:
        raise StopIteration
```

dot product

```python
def __matmul__(self, other):
    return self.x * other.x + self.y * other.y
```

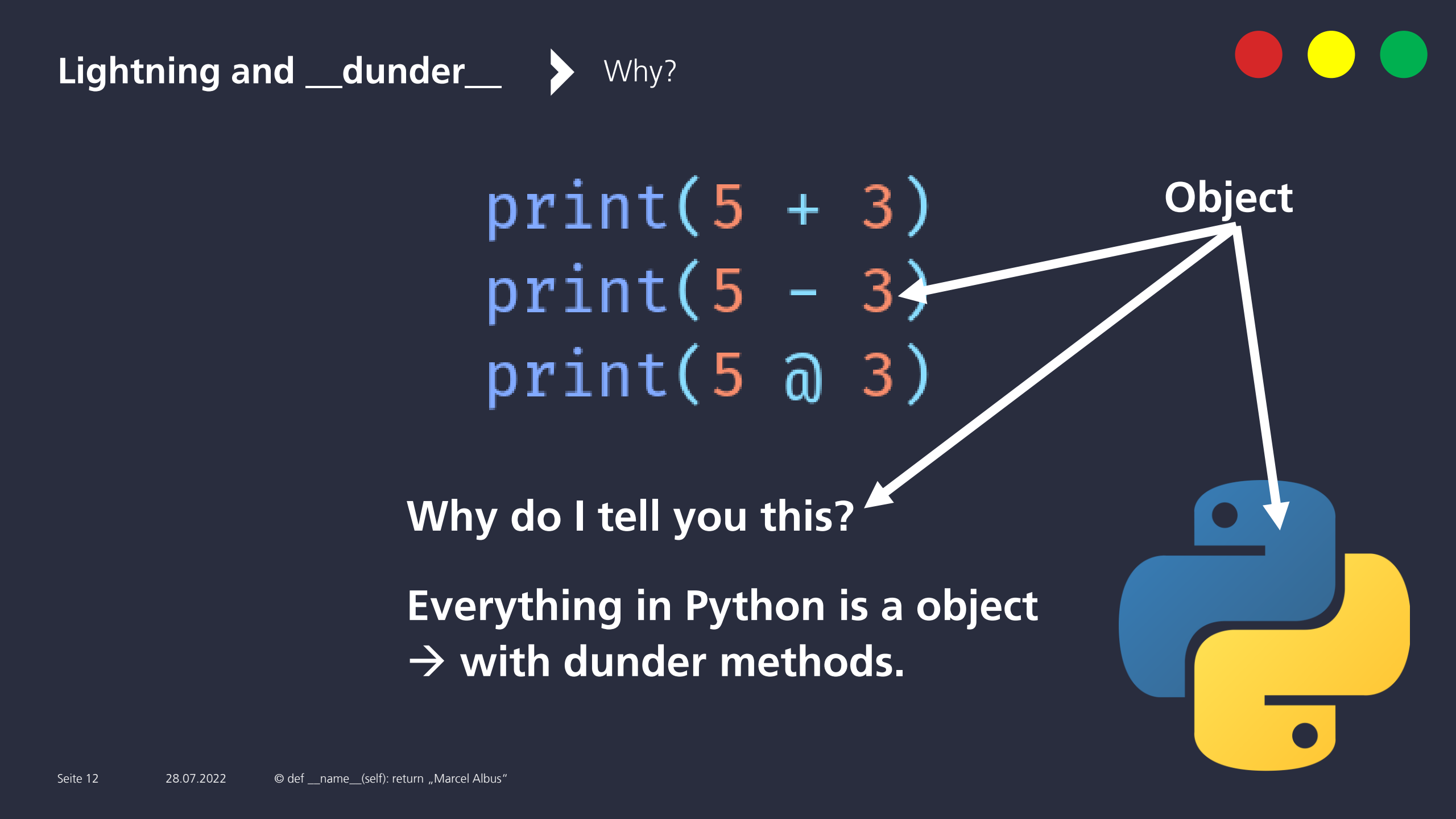
```python
@property
def len(self):
    return self.__matmul__(self) ** 0.5
```

sqrt of dot product with itself

Lightning and __dunder__ > What?

```python
print(Vector(2, 3) + Vector(5, 6))
# >> Vector[x = 7, y = 9]

for i in Vector(6, 7):
    print(i)
# >> 6
# >> 7

print(Vector(6, 7) @ Vector(2, 3))
# >> 33

print(Vector(6, 7).len)
# >> 9.219544457292887
```

Lightning and __dunder__ > Why?



```
print(5 + 3)
print(5 - 3)
print(5 @ 3)
```

Object

Why do I tell you this?

Everything in Python is a object
→ with dunder methods.

Lightning and __dunder__ > __del__(self)

```
print(bytes.fromhex("7468616E6B20796F75").decode("utf-8"))
# >> thank you
```

