Comprehending Comprehensions

by Rodrigo Girão Serrão

Rodrigo 🐍🚀

@mathsppblog

Rodrigo 🐍🚀

@mathsppblog

Textualize

mathspp.com

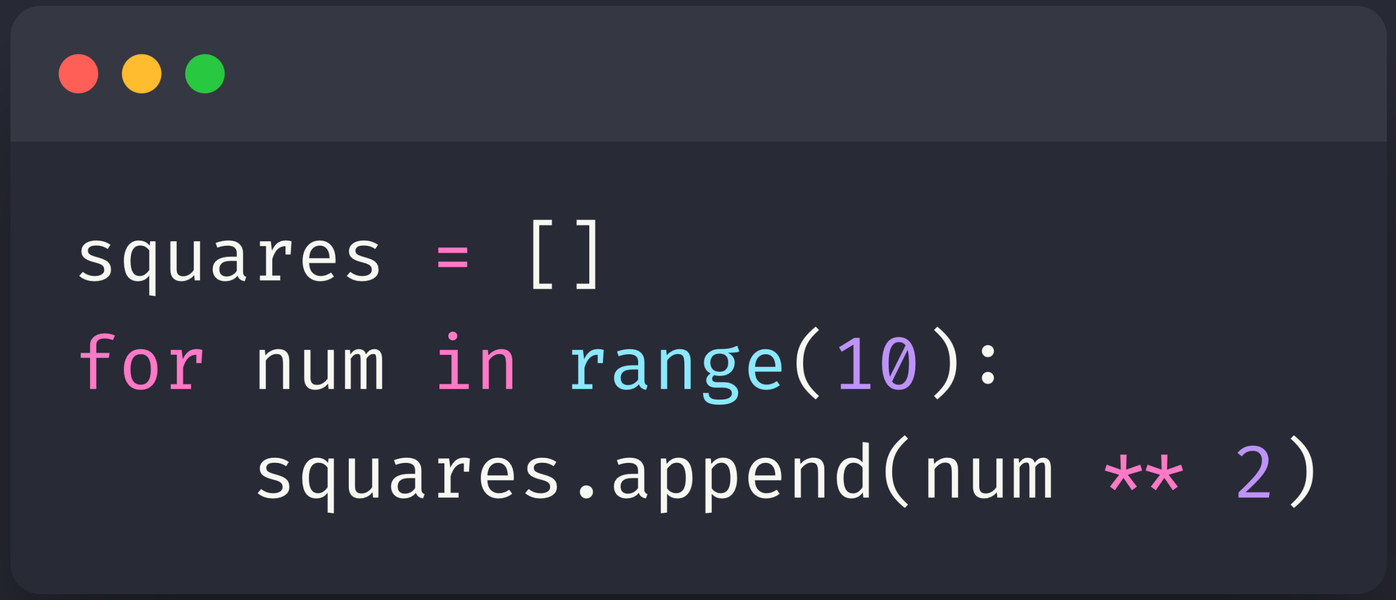
What's a (list) comprehension?

It's a way to build lists.

rodrigo@mathspp.com

Better understood
when compared to a loop.

rodrigo@mathspp.com



```
squares = []
for num in range(10):
    squares.append(num ** 2)
```

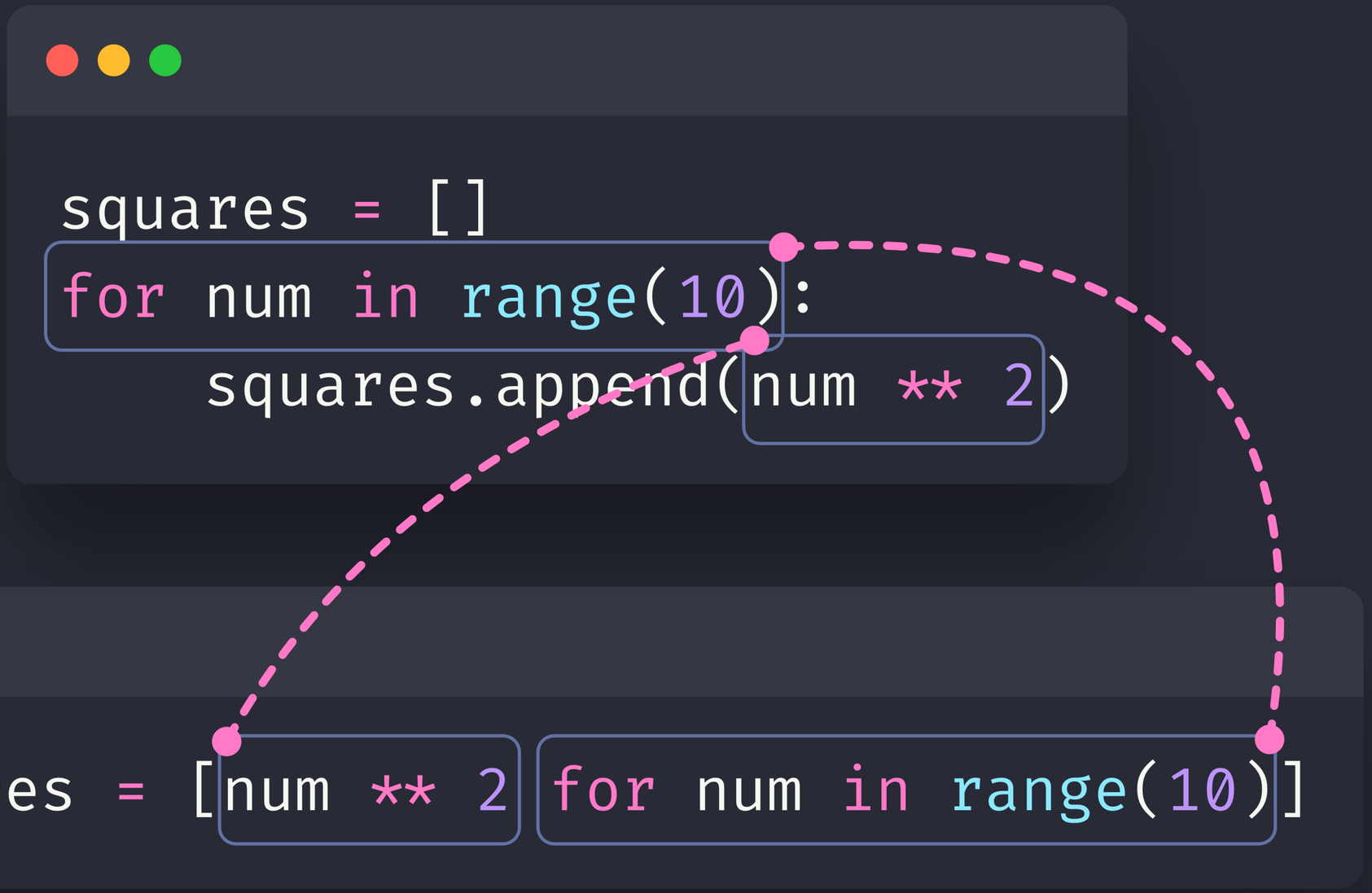
rodrigo@mathspp.com

```python
squares = []
for num in range(10):
    squares.append(num ** 2)
```

```python
squares = [num ** 2 for num in range(10)]
```

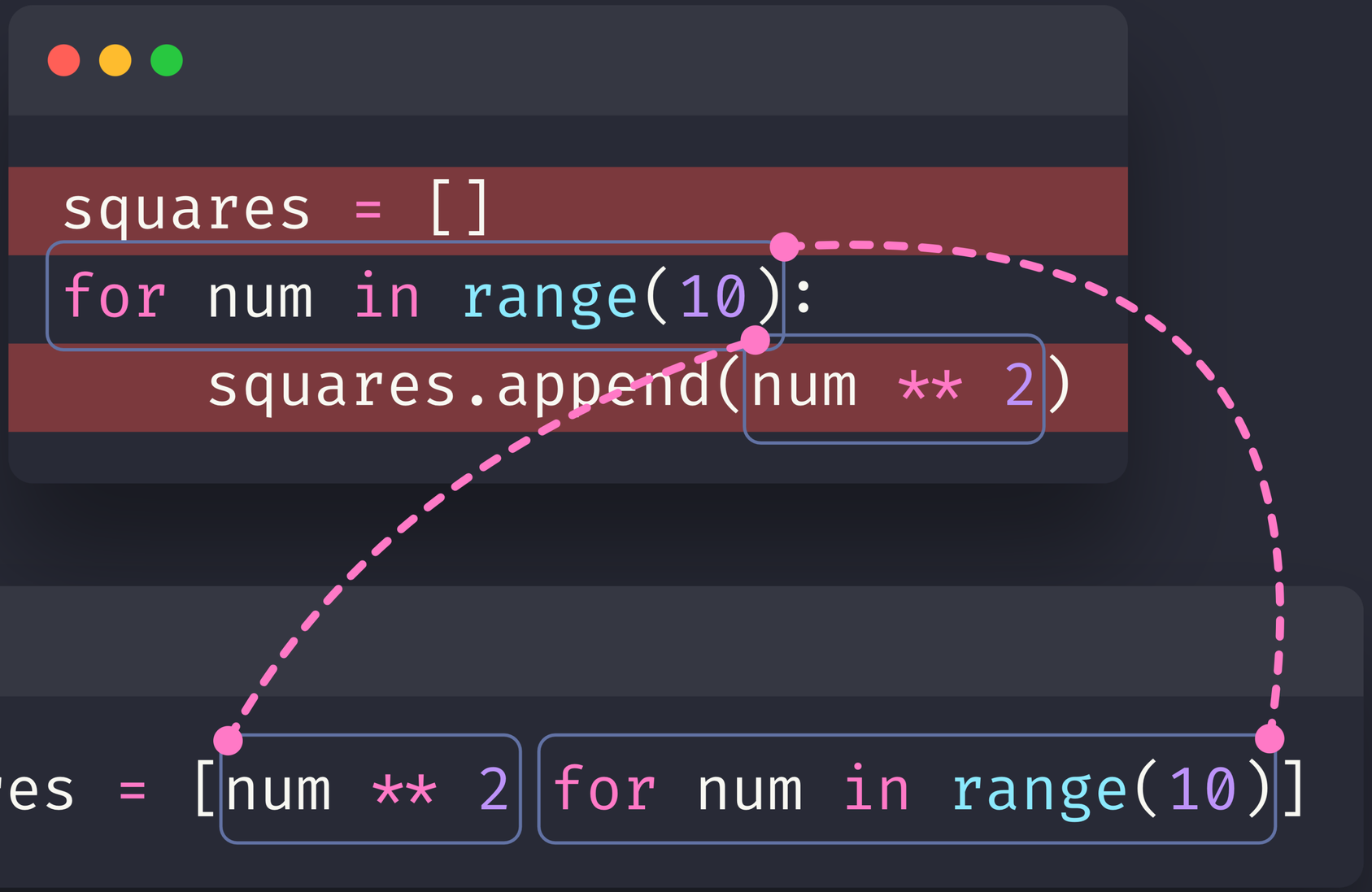
rodrigo@mathspp.com

Extract important parts.



rodrigo@mathspp.com

Drop fluff.



rodrigo@mathspp.com

How to write ***ANY*** (list) comprehension:

rodrigo@mathspp.com

How to write ***ANY*** (list) comprehension:

```
new_list = []
for element in iterable:
    new_list.append(func(element))
```

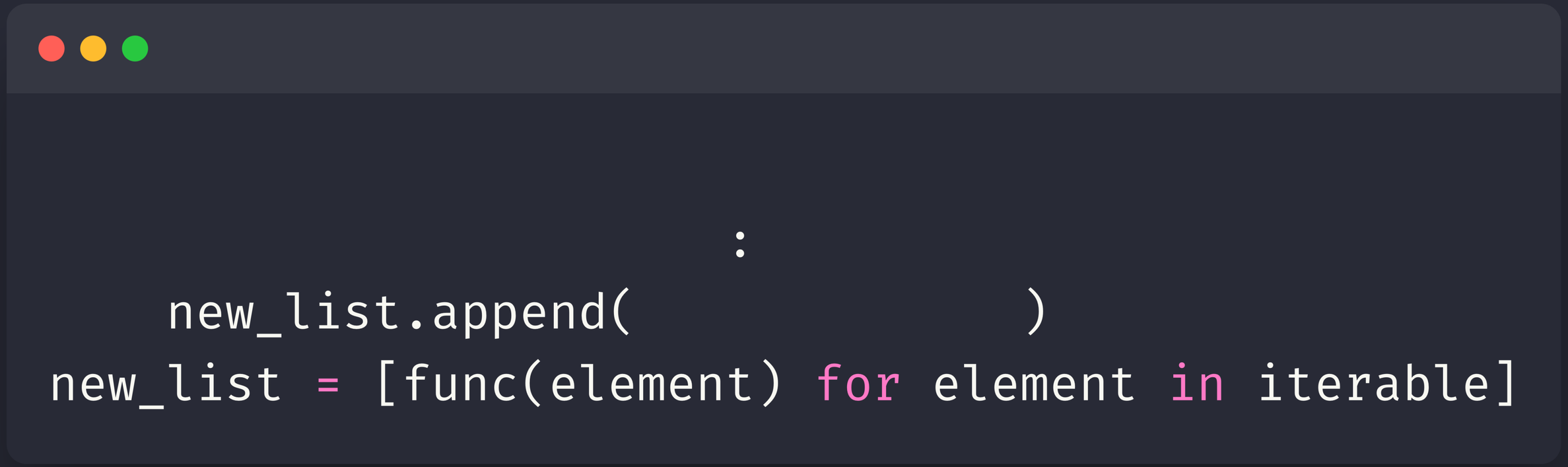
rodrigo@mathspp.com

```
for element in iterable:
    new_list.append(func(element))
new_list = [                      ]
```

rodrigo@mathspp.com

```
for element in iterable:
    new_list.append(                )
new_list = [func(element)                ]
```

rodrigo@mathspp.com



rodrigo@mathspp.com

```
                                :
    new_list.append(                    )
new_list = [func(element) for element in iterable]
```

rodrigo@mathspp.com

Filtering data

```python
odd_squares = []

for num in range(10):
    if num % 2:
        odd_squares.append(num ** 2)
```

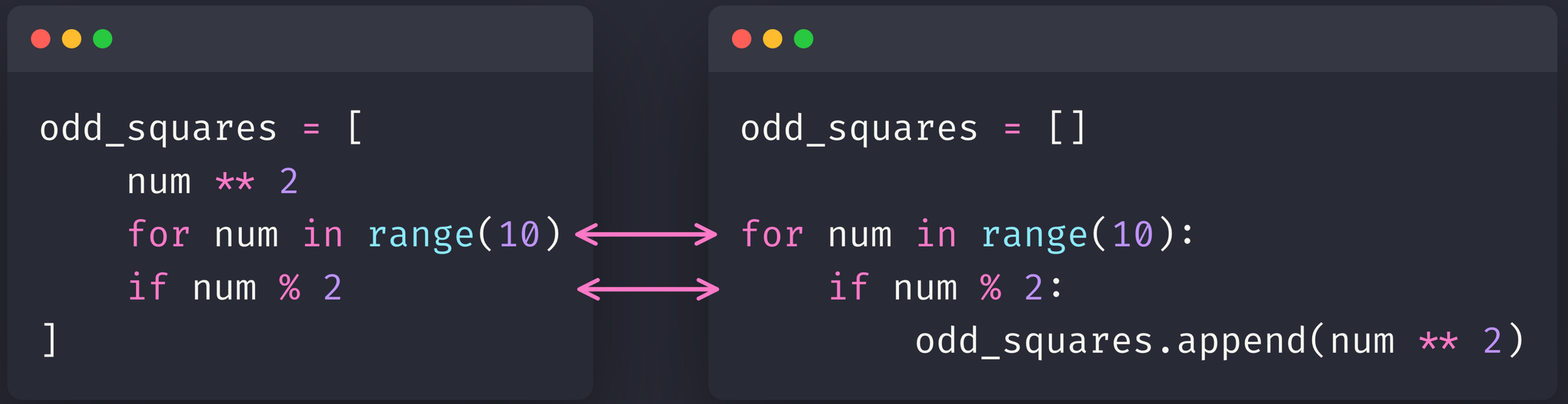
rodrigo@mathspp.com

```python
odd_squares = [
    num ** 2
    for num in range(10)
    if num % 2
]
```

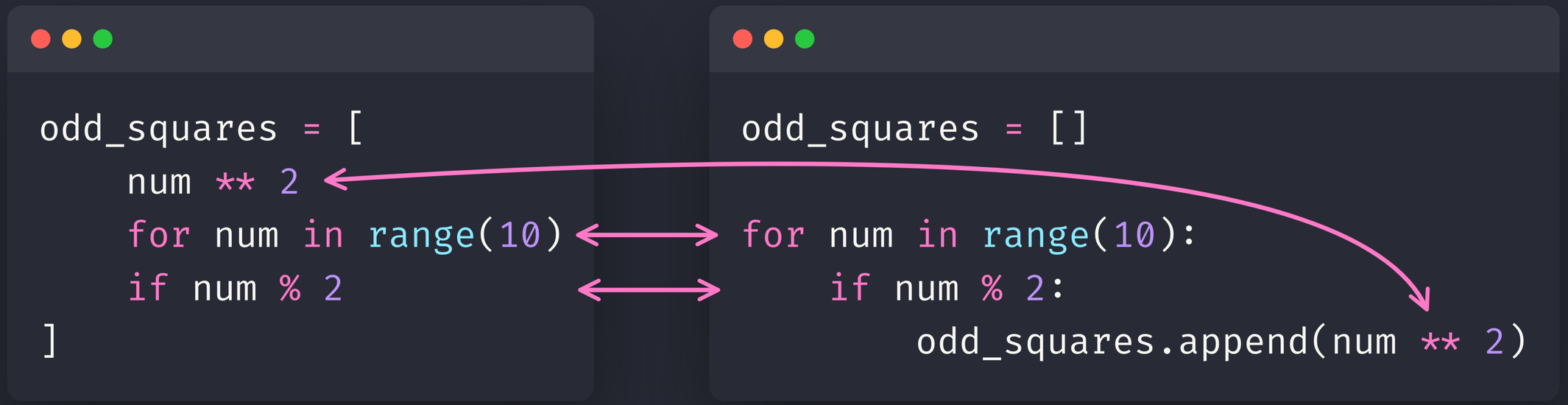
```python
odd_squares = []

for num in range(10):
    if num % 2:
        odd_squares.append(num ** 2)
```

rodrigo@mathspp.com



rodrigo@mathspp.com



rodrigo@mathspp.com

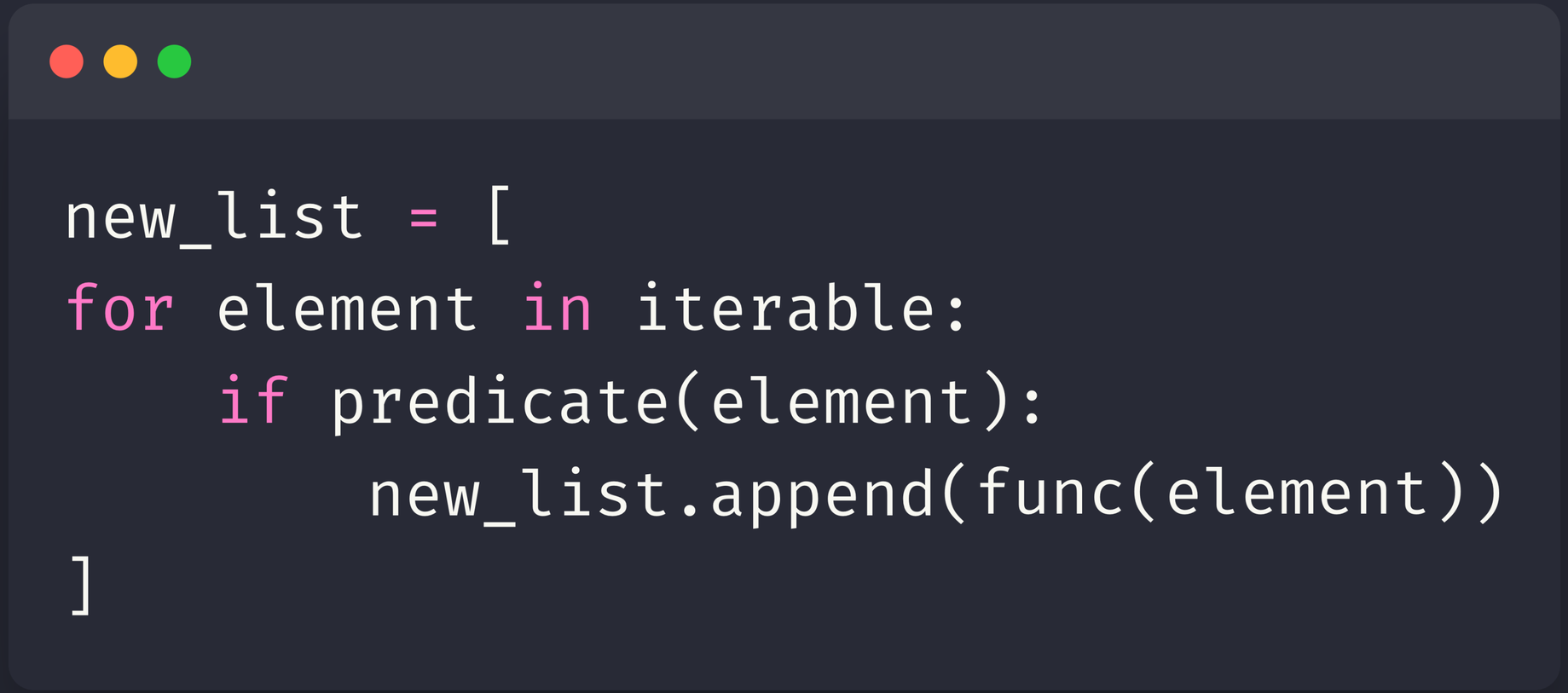
How to write ***ANY*** (list) comprehension (take 2):

rodrigo@mathspp.com

How to write ***ANY*** (list) comprehension (take 2):

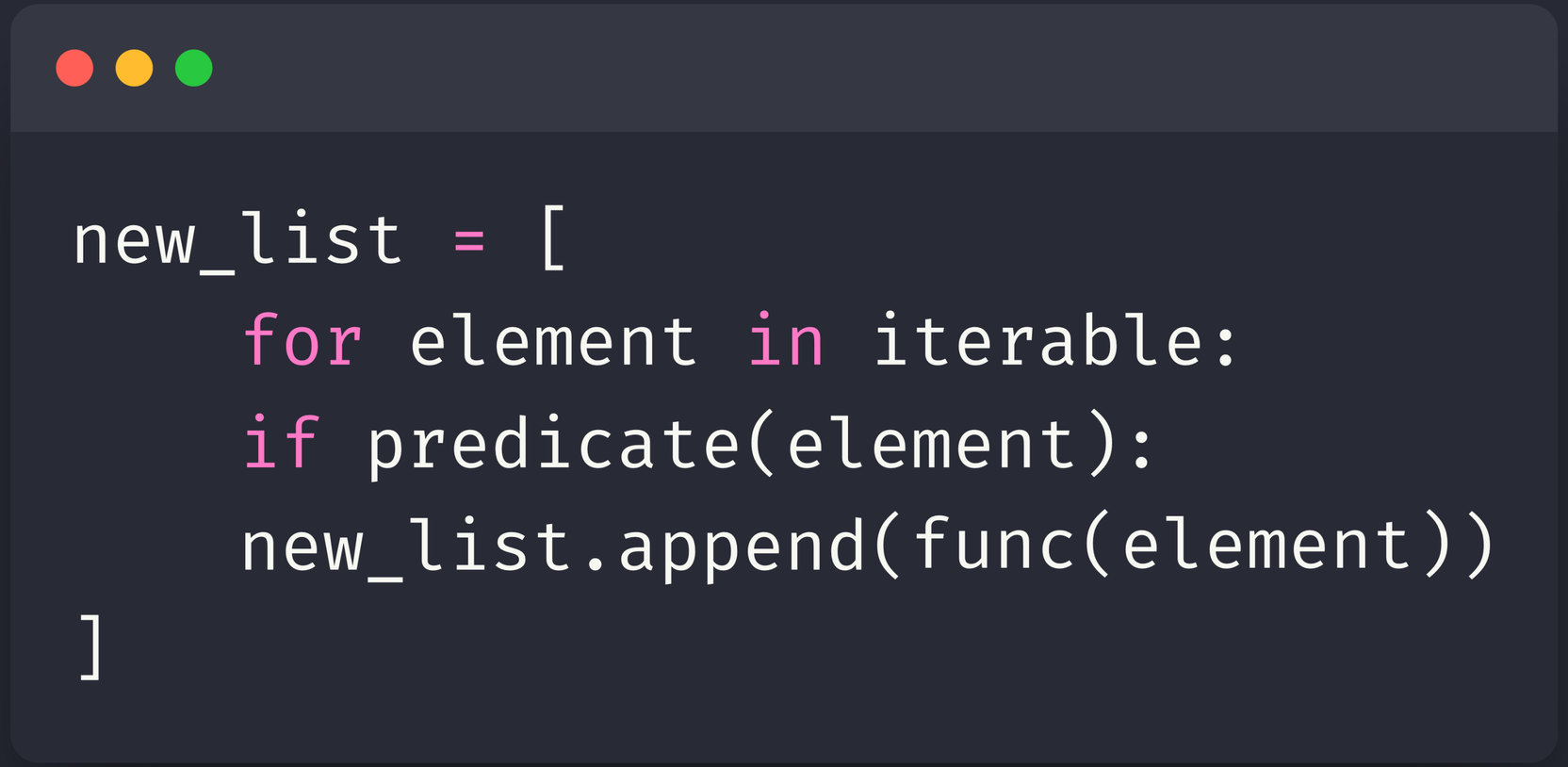
```
new_list = []
for element in iterable:
    if predicate(element):
        new_list.append(func(element))
```

rodrigo@mathspp.com



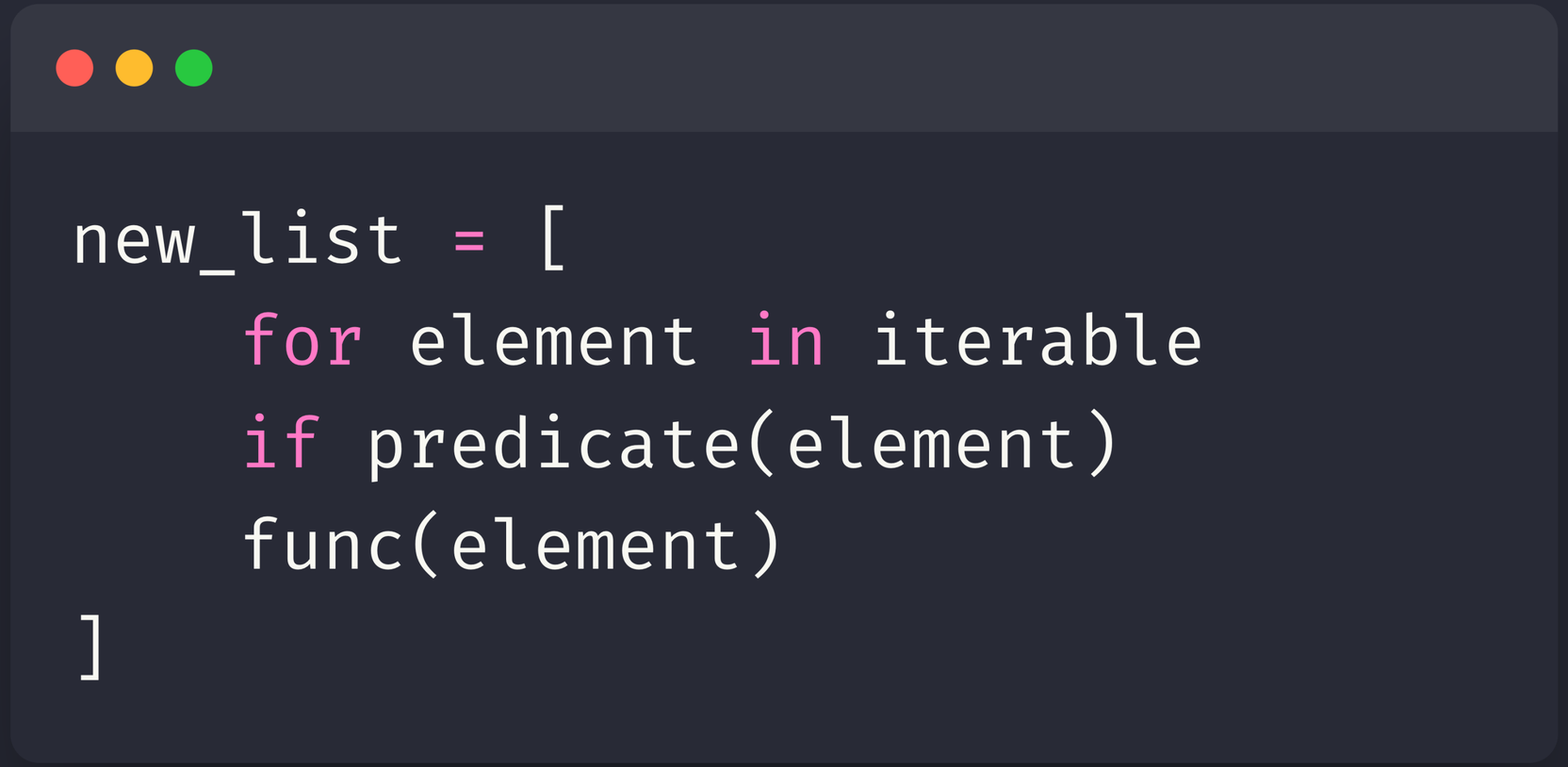
```
new_list = [
for element in iterable:
    if predicate(element):
        new_list.append(func(element))
]
```

rodrigo@mathspp.com



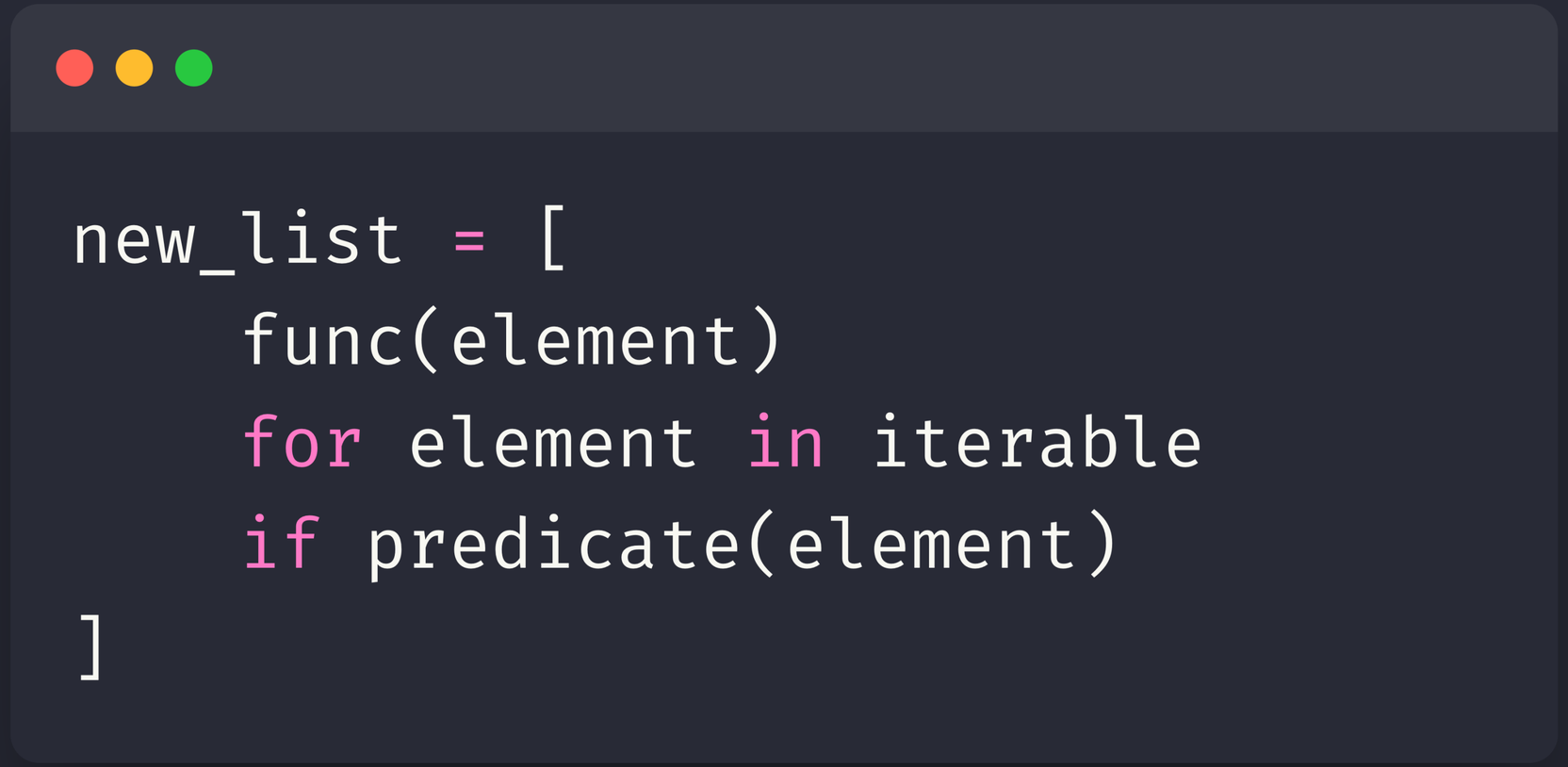
```
new_list = [
    for element in iterable:
    if predicate(element):
    new_list.append(func(element))
]
```

rodrigo@mathspp.com



```
new_list = [
    for element in iterable
    if predicate(element)
    func(element)
]
```

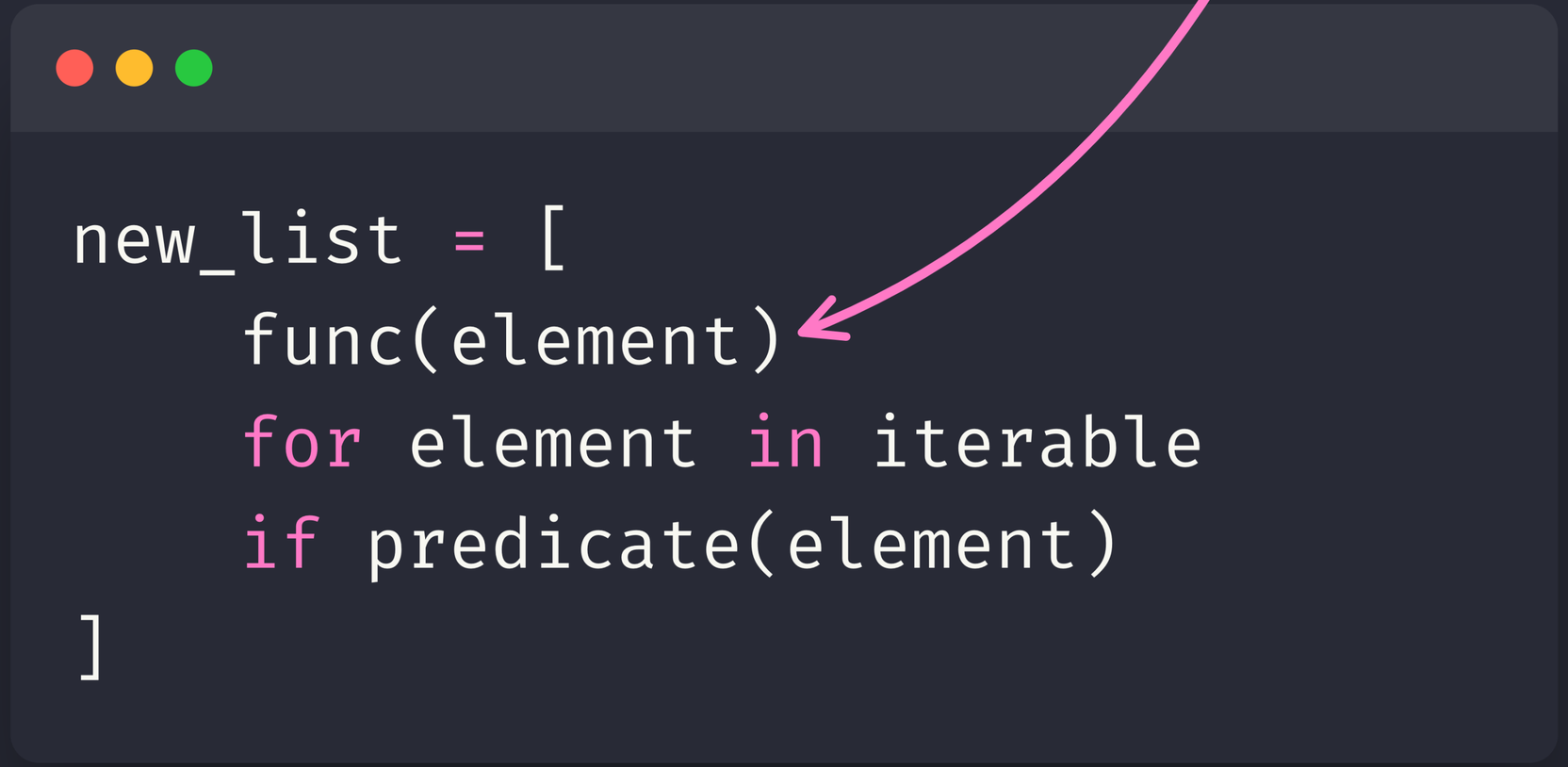
rodrigo@mathspp.com



```
new_list = [
    func(element)
    for element in iterable
    if predicate(element)
]
```

rodrigo@mathspp.com

1. Data transformation.



```
new_list = [
    func(element)
    for element in iterable
    if predicate(element)
]
```

rodrigo@mathspp.com

```
new_list = [
    func(element)
    for element in iterable
    if predicate(element)
]
```

2. Data source.

rodrigo@mathspp.com

```
new_list = [
    func(element)
    for element in iterable
    if predicate(element)
]
```

3. Data filter (optional).

rodrigo@mathspp.com

That's all there is to it.

rodrigo@mathspp.com

That's all there is to it.

Really.

rodrigo@mathspp.com

Why bother?

Fast(er) ⚡

rodrigo@mathspp.com

Short(er) 👈

rodrigo@mathspp.com

Pur(er) ✨

rodrigo@mathspp.com

Fast(er) ⚡

Short(er) 🤏

Pur(er) ✨

rodrigo@mathspp.com

rodrigo@mathspp.com

rodrigo@mathspp.com

*English is hard!

rodrigo@mathspp.com

Readability was subjective..?

rodrigo@mathspp.com

```
my_list = []
for it2 in it1:
 for _ in it3:
  if p1(it2):
   for it4 in it2:
    for v1 in it4:
     if p2(v1):
      if p3(it4):
       for it6 in it5:
        for v2, it7 in it6:
         for v3, it8, it9 in it7:
          if p2(v1):
           for v4, v5 in zip(it8, it9):
            my_list.append(func(v1, v2, v3, v4, v5))
```

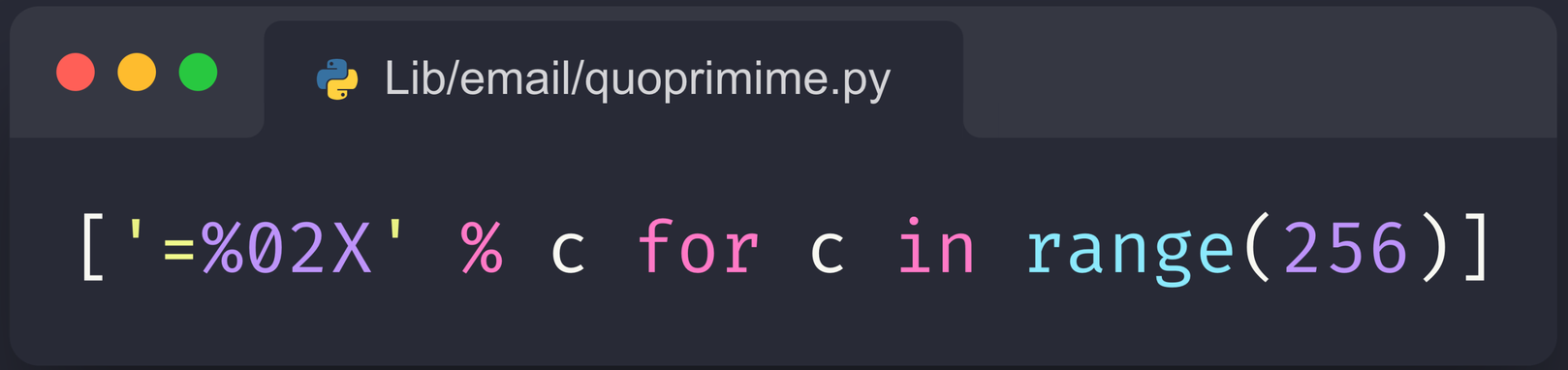
rodrigo@mathspp.com

```
my_list = [
 func(v1, v2, v3, v4, v5)
 for it2 in it1
 for _ in it3
 if p1(it2)
 for it4 in it2
 for v1 in it4
 if p2(v1)
 if p3(it4)
 for it6 in it5
 for v2, it7 in it6
 for v3, it8, it9 in it7
 if p2(v1)
 for v4, v5 in zip(it8, it9)
]
```

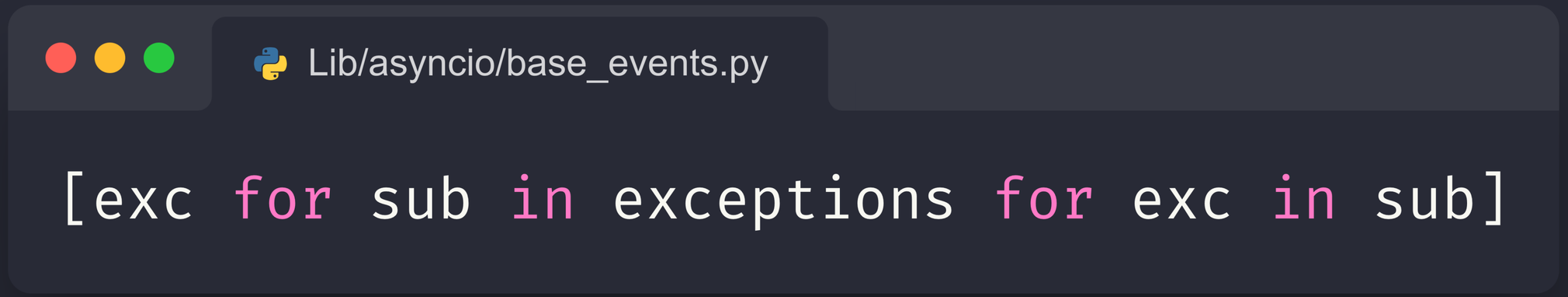
```
my_list = []
for it2 in it1:
 for _ in it3:
  if p1(it2):
   for it4 in it2:
    for v1 in it4:
     if p2(v1):
      if p3(it4):
       for it6 in it5:
        for v2, it7 in it6:
         for v3, it8, it9 in it7:
          if p2(v1):
           for v4, v5 in zip(it8, it9):
            my_list.append(func(v1, v2, v3, v4, v5))
```

rodrigo@mathspp.com

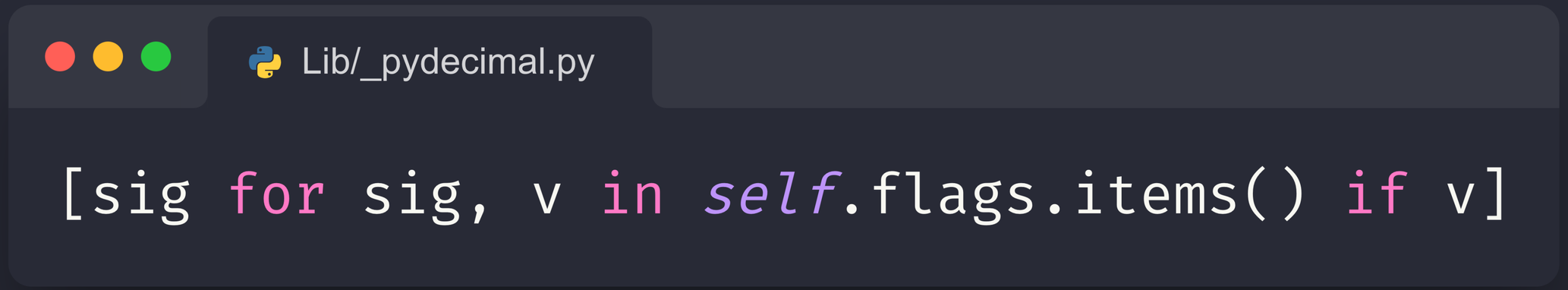
Examples ✨



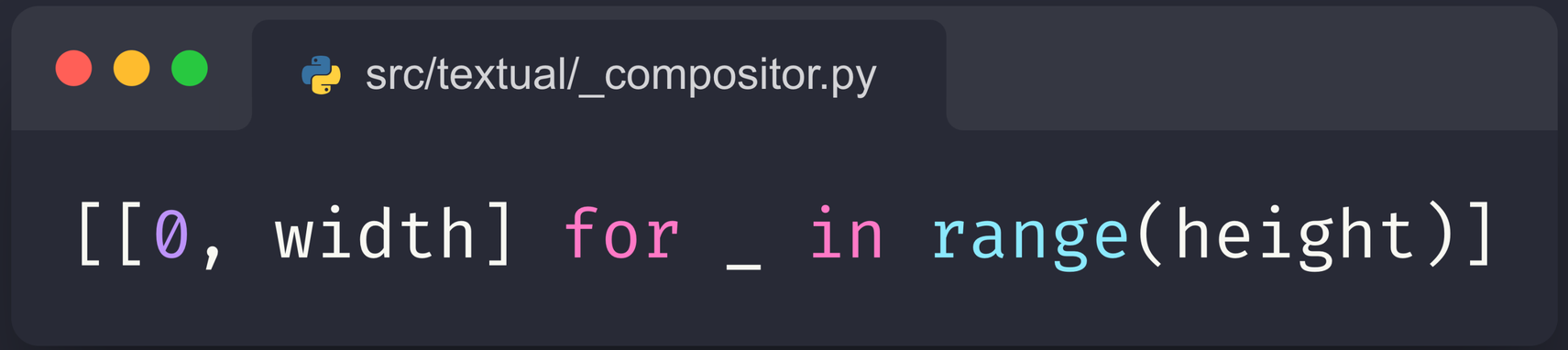
rodrigo@mathspp.com



rodrigo@mathspp.com



rodrigo@mathspp.com



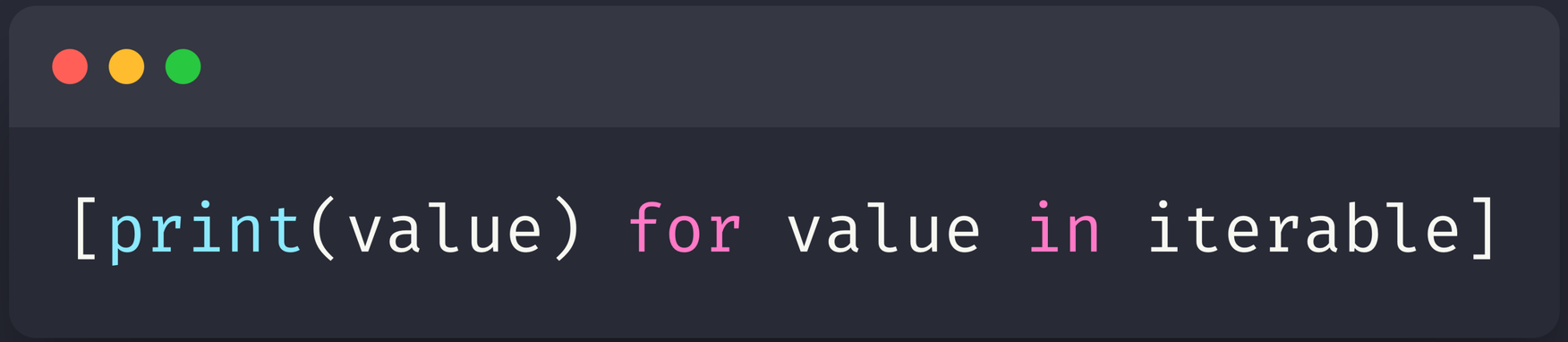
rodrigo@mathspp.com

src/textual/css/stylesheet.py

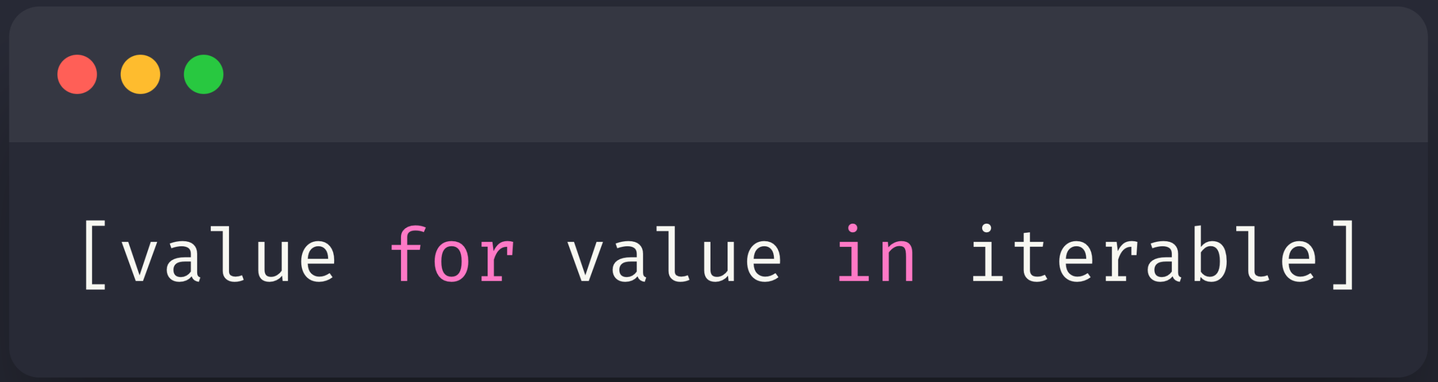
```
[
    rule
    for rule in reversed(self.rules)
    if rule in limit_rules
]
```

rodrigo@mathspp.com

Examples 🤢

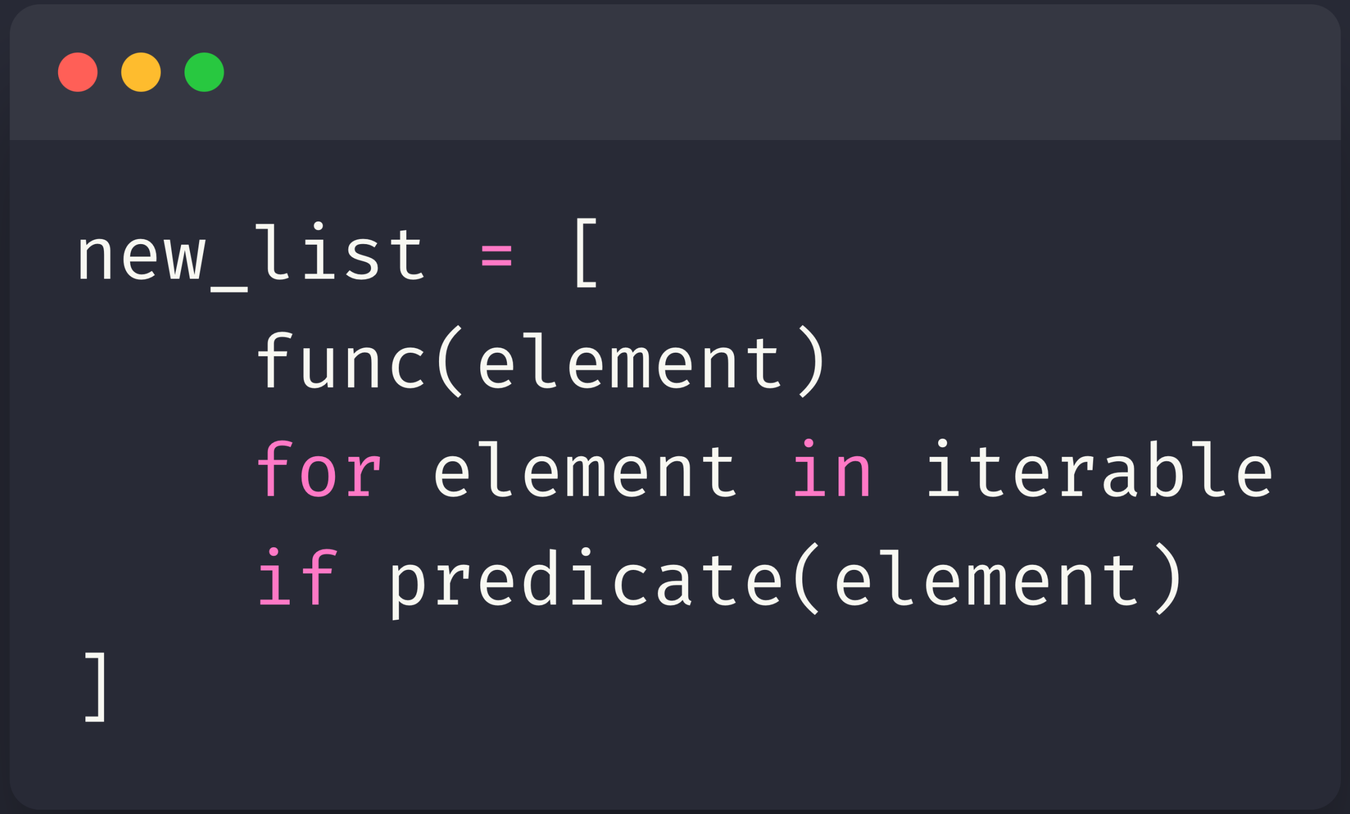


rodrigo@mathspp.com



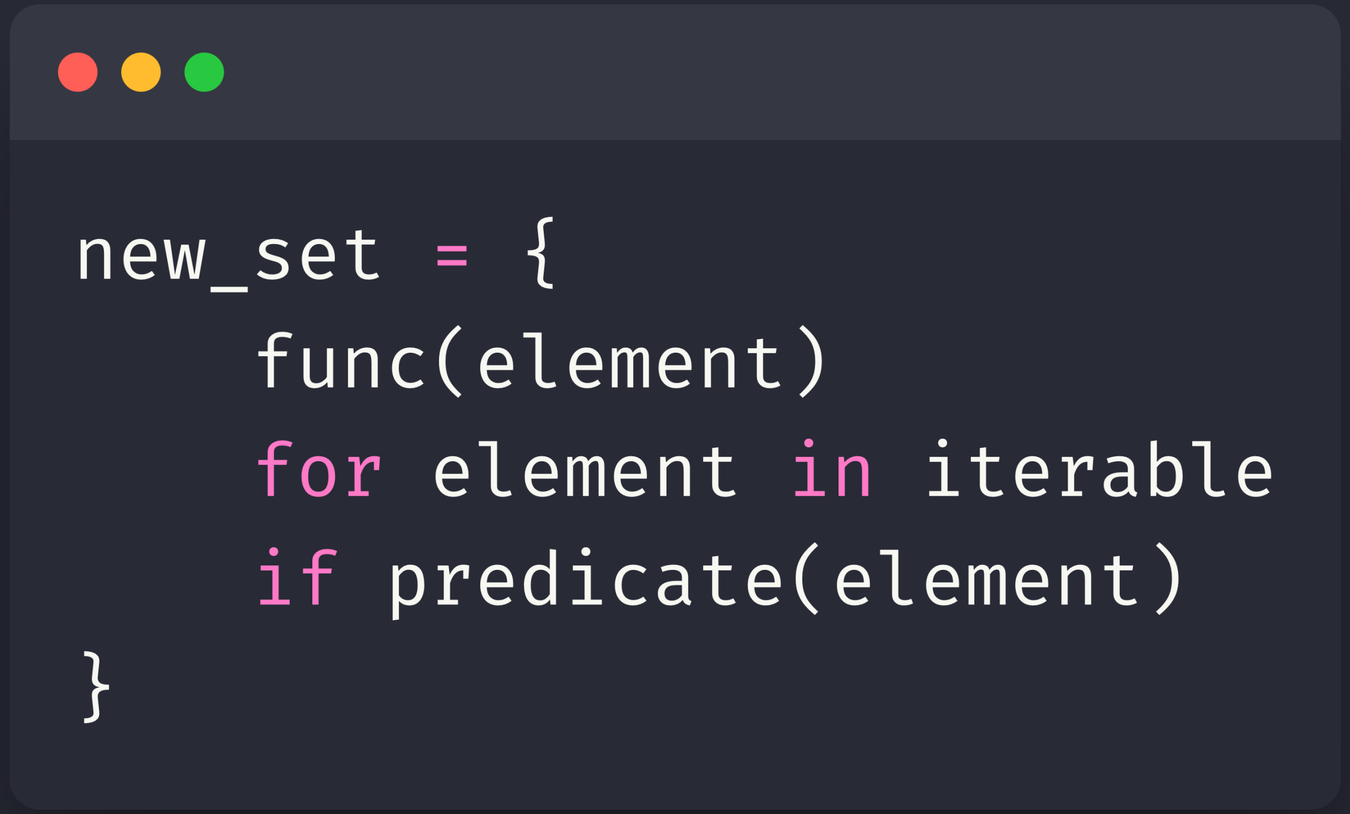
rodrigo@mathspp.com

Set comprehensions



```
new_list = [
    func(element)
    for element in iterable
    if predicate(element)
]
```

rodrigo@mathspp.com



```
new_set = {
    func(element)
    for element in iterable
    if predicate(element)
}
```

rodrigo@mathspp.com

That's all there is to it.

rodrigo@mathspp.com

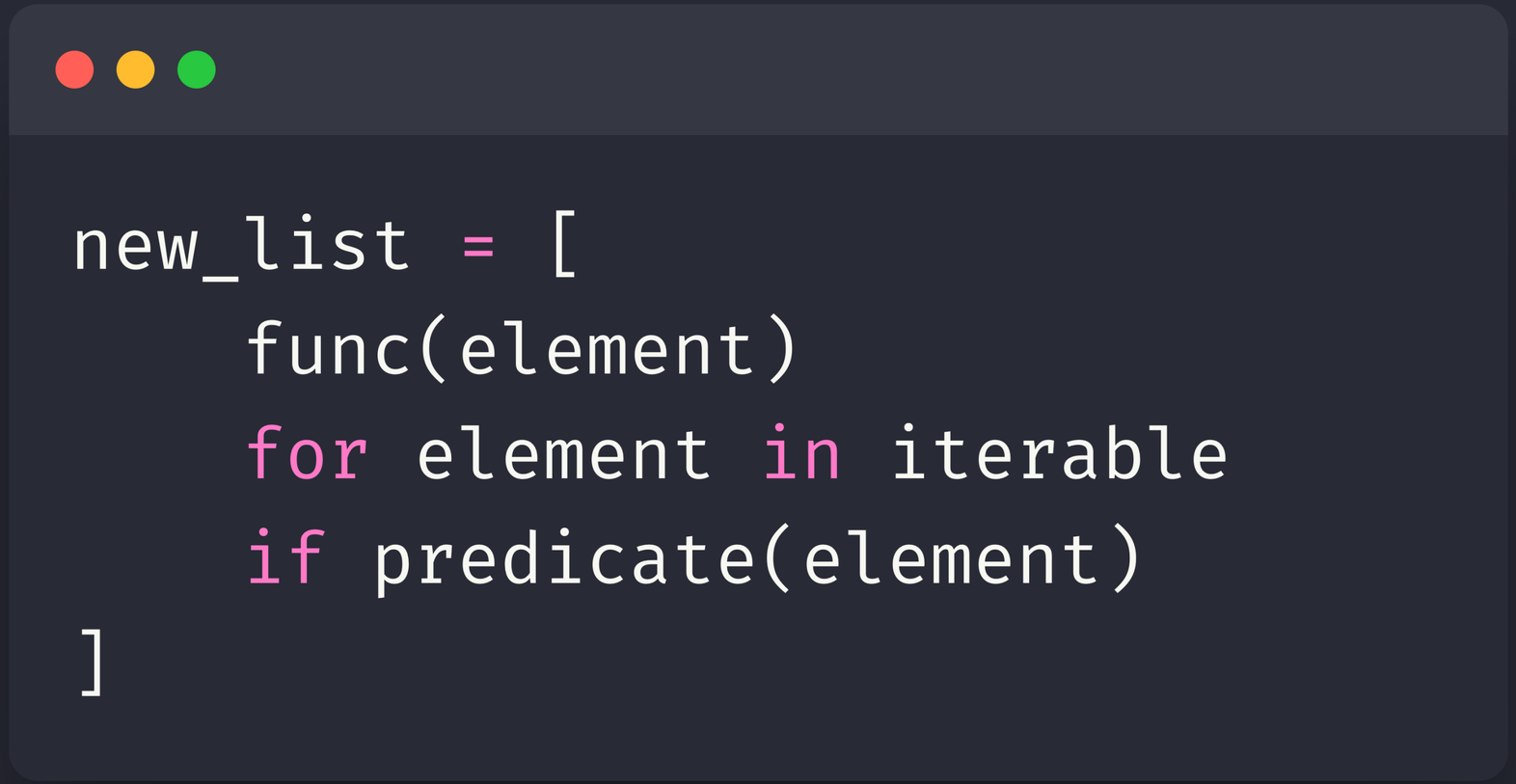
That's all there is to it.

Really.

rodrigo@mathspp.com

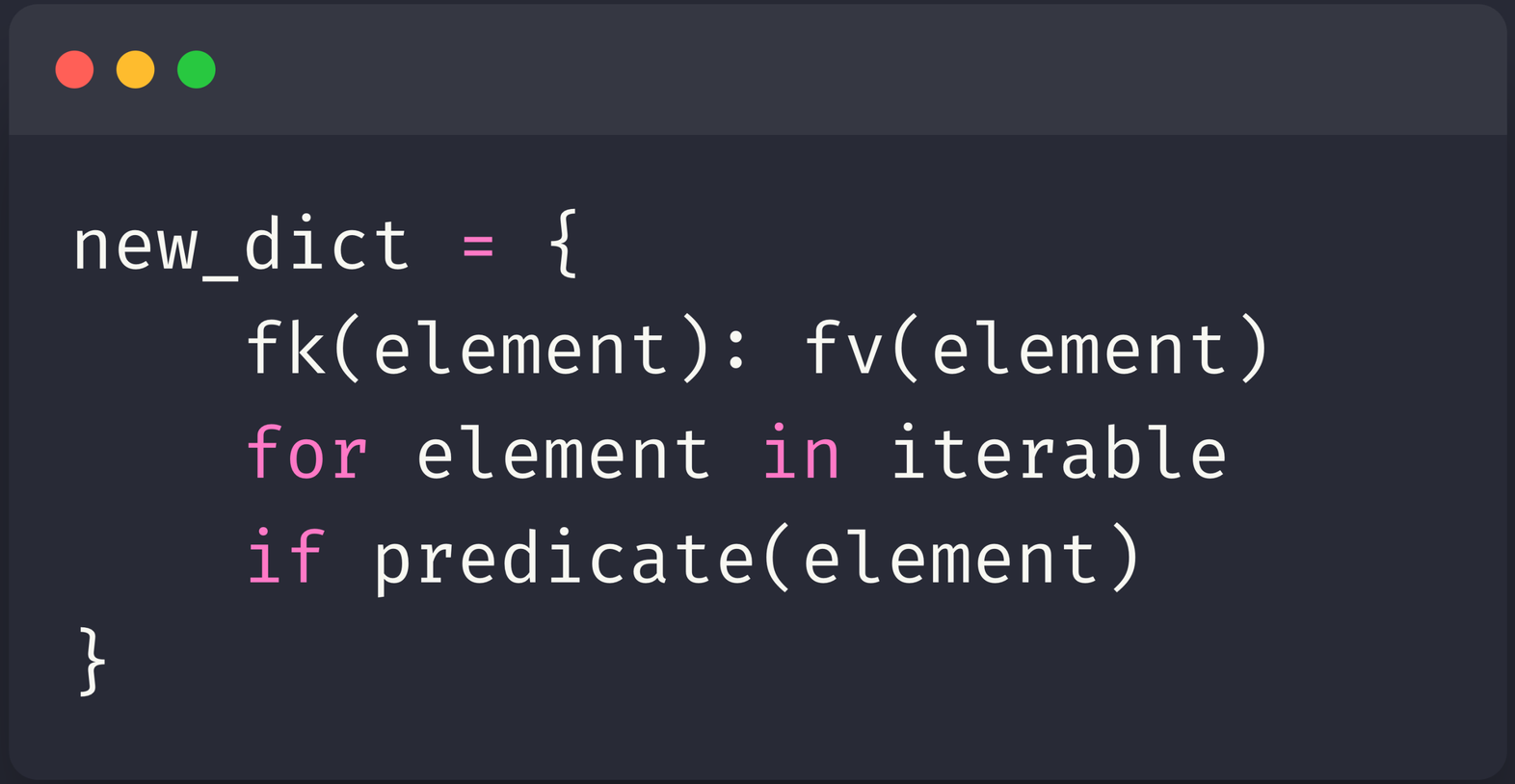
rodrigo@mathspp.com

Dictionary comprehensions



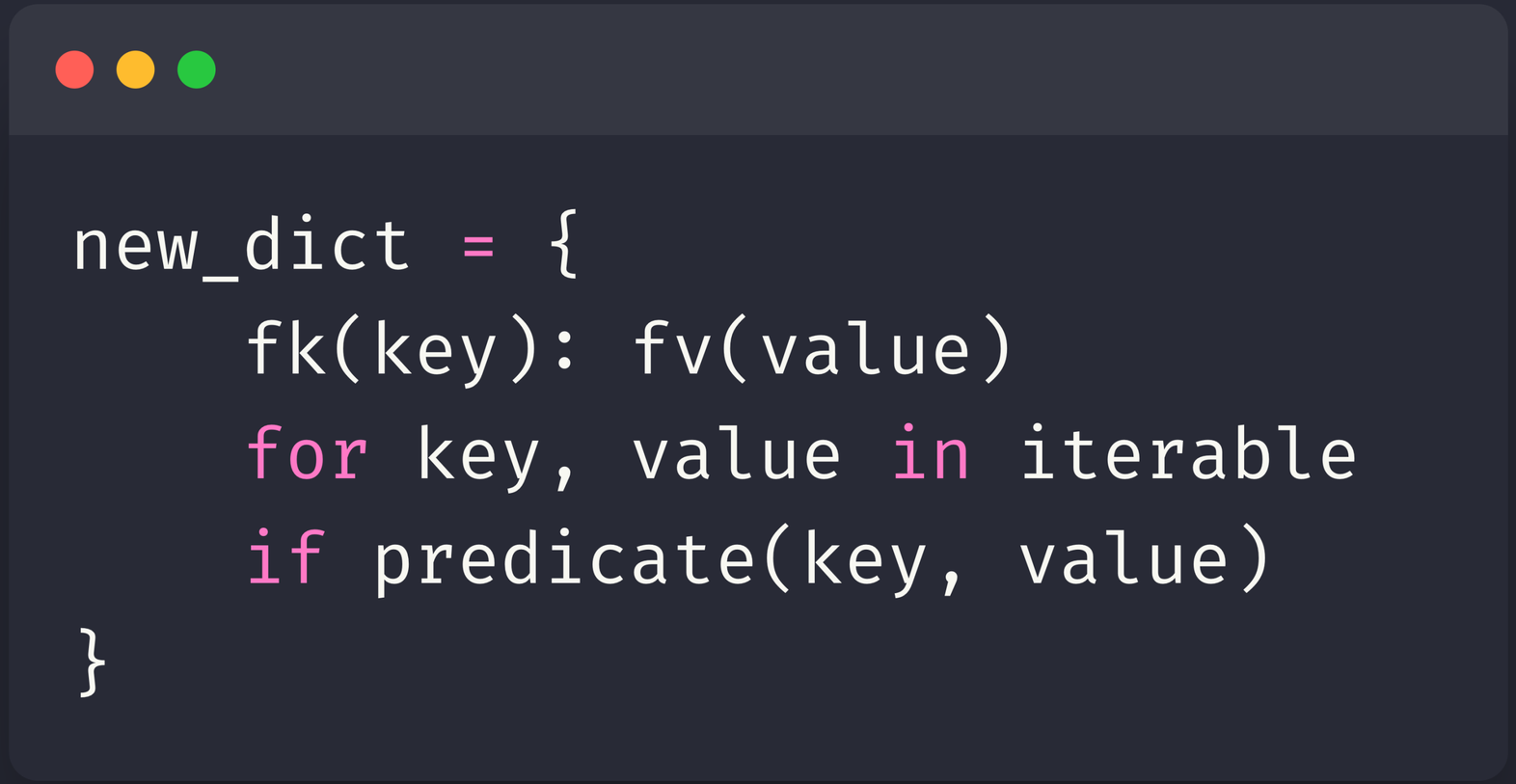
```
new_list = [
    func(element)
    for element in iterable
    if predicate(element)
]
```

rodrigo@mathspp.com



```
new_dict = {
    fk(element): fv(element)
    for element in iterable
    if predicate(element)
}
```

rodrigo@mathspp.com



```
new_dict = {
    fk(key): fv(value)
    for key, value in iterable
    if predicate(key, value)
}
```

rodrigo@mathspp.com

That's all there is to it.

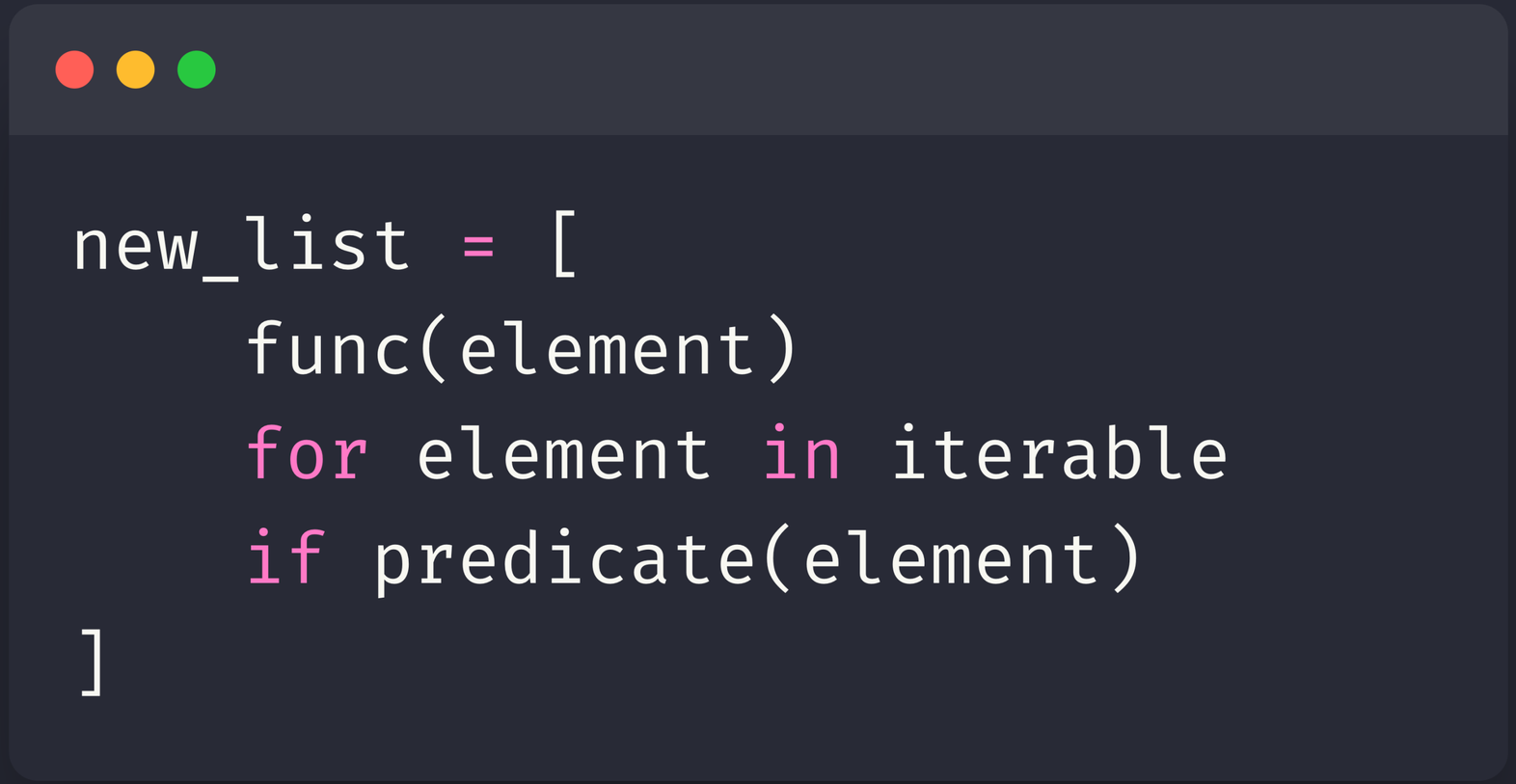
rodrigo@mathspp.com

That's all there is to it.

Really.

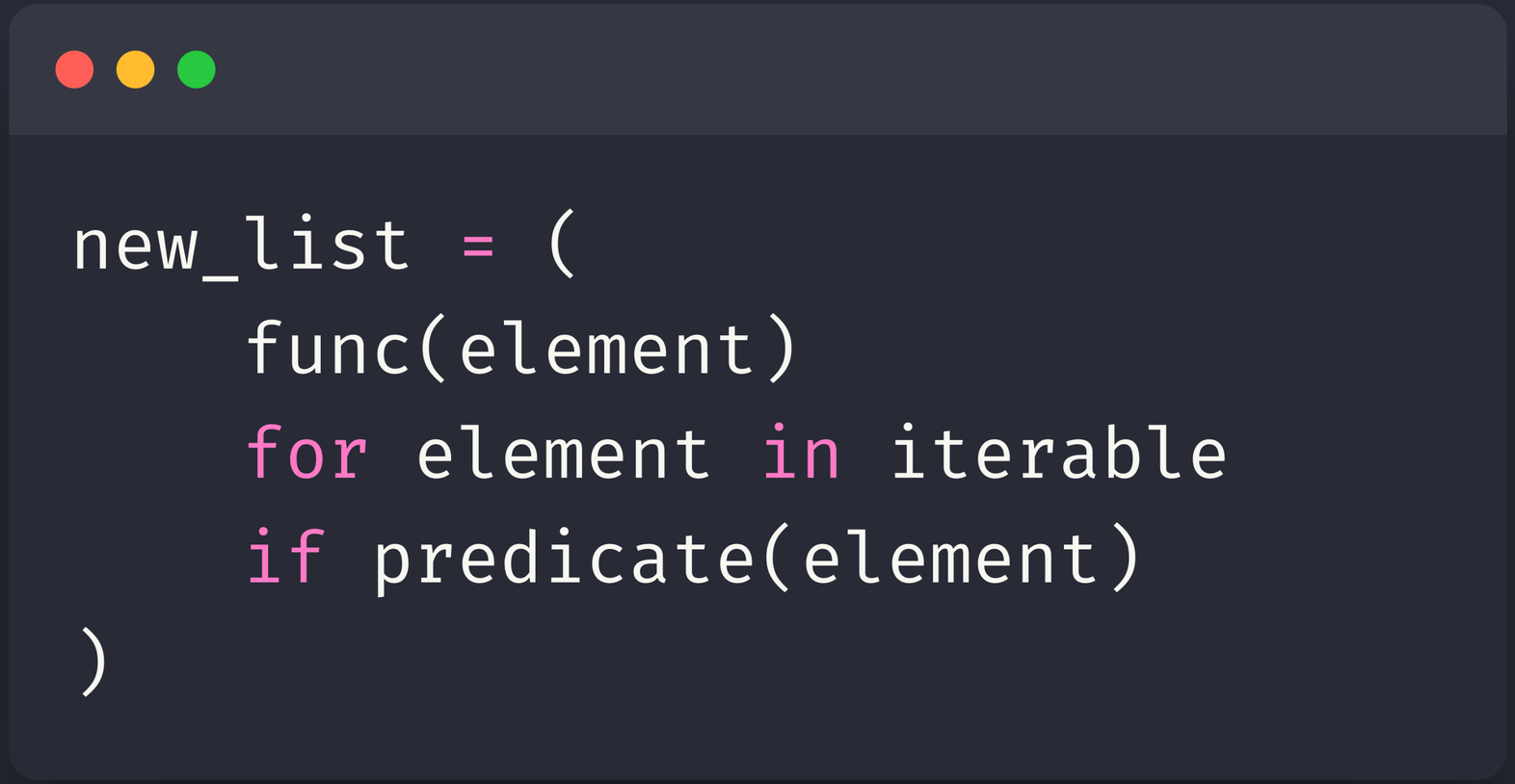
rodrigo@mathspp.com

Generator expressions



```
new_list = [
    func(element)
    for element in iterable
    if predicate(element)
]
```

rodrigo@mathspp.com



```
new_list = (
    func(element)
    for element in iterable
    if predicate(element)
)
```

rodrigo@mathspp.com

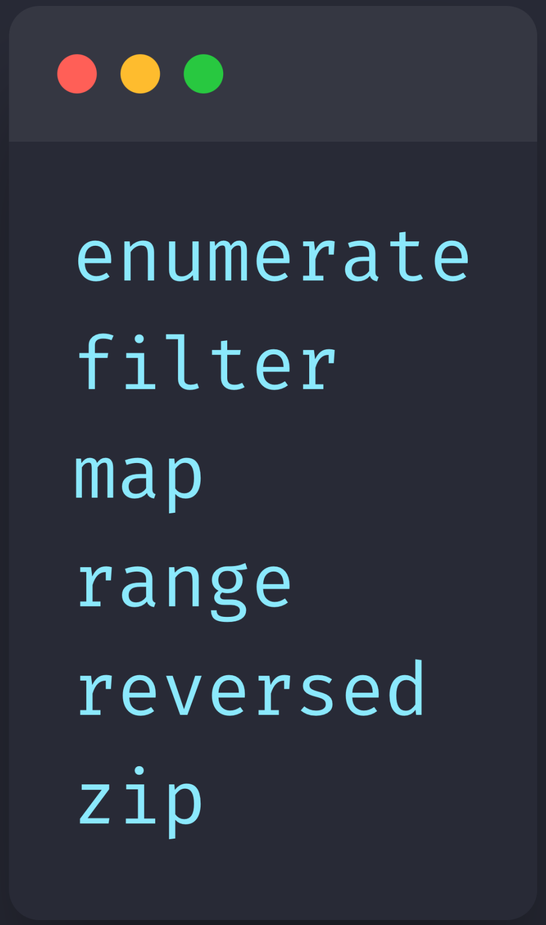
That's all there is to it.

rodrigo@mathspp.com

That's all there is to it.

Really.

rodrigo@mathspp.com



rodrigo@mathspp.com

Workshop tomorrow.

rodrigo@mathspp.com

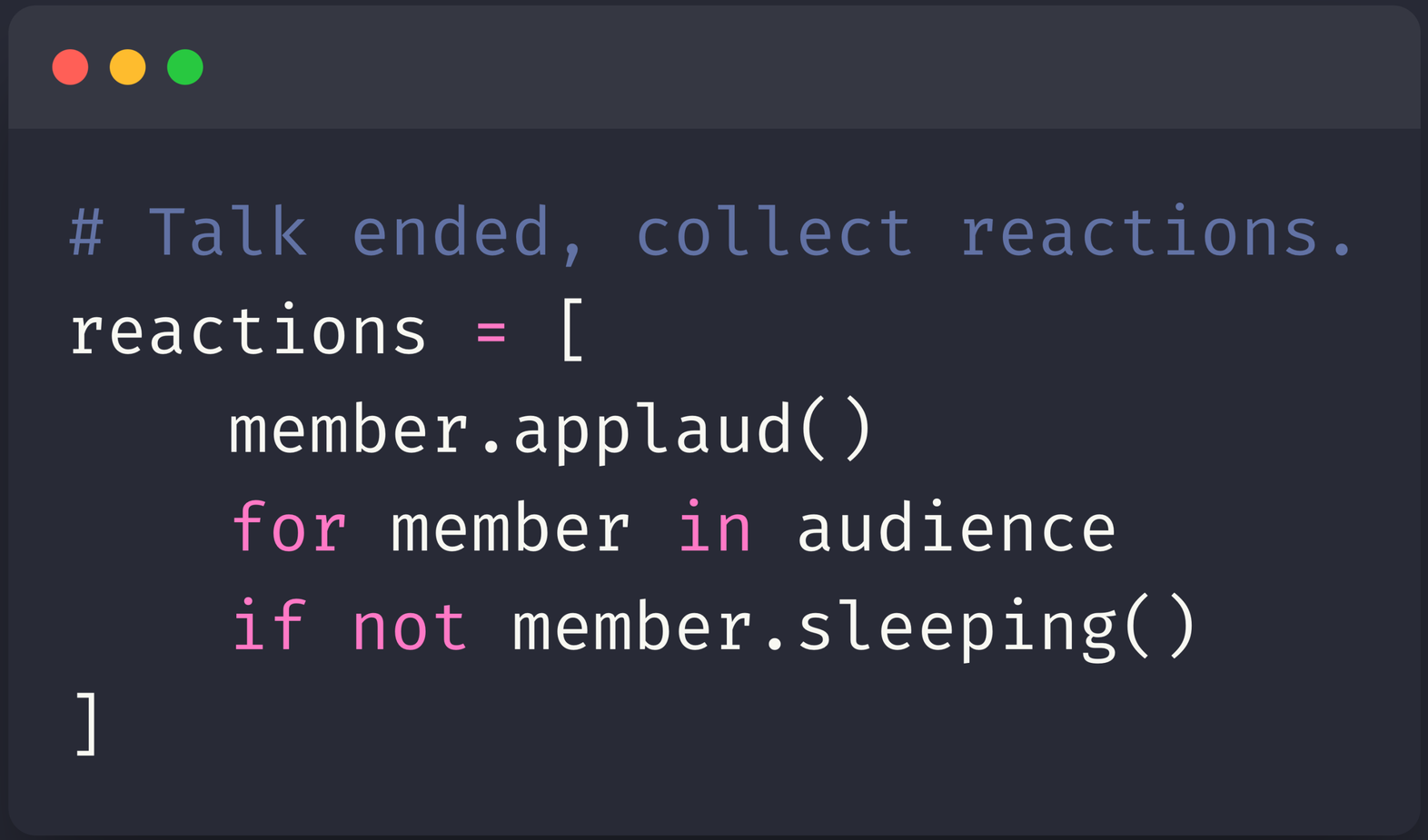
rodrigo@mathspp.com

Just practice

Workshop tomorrow!

rodrigo@mathspp.com

Workshop tomorrow!



```
# Talk ended, collect reactions.
reactions = [
    member.applaud()
    for member in audience
    if not member.sleeping()
]
```

rodrigo@mathspp.com

presented with  snappify*

*no affiliation, they're just awesome

rodrigo@mathspp.com

rodrigo@mathspp.com

mathspp.com/talks