```
Given my name is Ilja and I'm 24 years old
Then assert the given person is 24 years old

# and I‘m working for cuescience
```

behave

```python
@given("my name is {name:s} and I'm {age:d} years old")
def my_name_and_age(context, name, age):
    context.person = Person(name, age)
```

behave

```python
@then("assert the given person is {age:d} years old")
def assert_person_is_n_years_old(context, age):
    assert context.person.age == age
```

behave

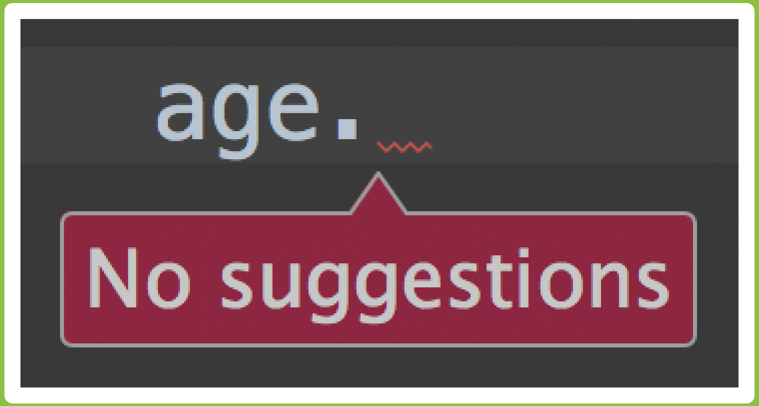
```
Feature: Test  # test.feature:1
Scenario: Test # test.feature:2
Given my name is Ilja and I'm 24 years old # steps/steps.py:10 0.000s
Then assert the given person is 24 years old # steps/steps.py:6 0.000s

1 feature passed, 0 failed, 0 skipped
1 scenario passed, 0 failed, 0 skipped
2 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.000s
```

behave

```python
@given("my name is {name:s} and I'm {age:d} years old")
def my_name_and_age(context, name, age):
    context.person = Person(name, age)
```

code completion

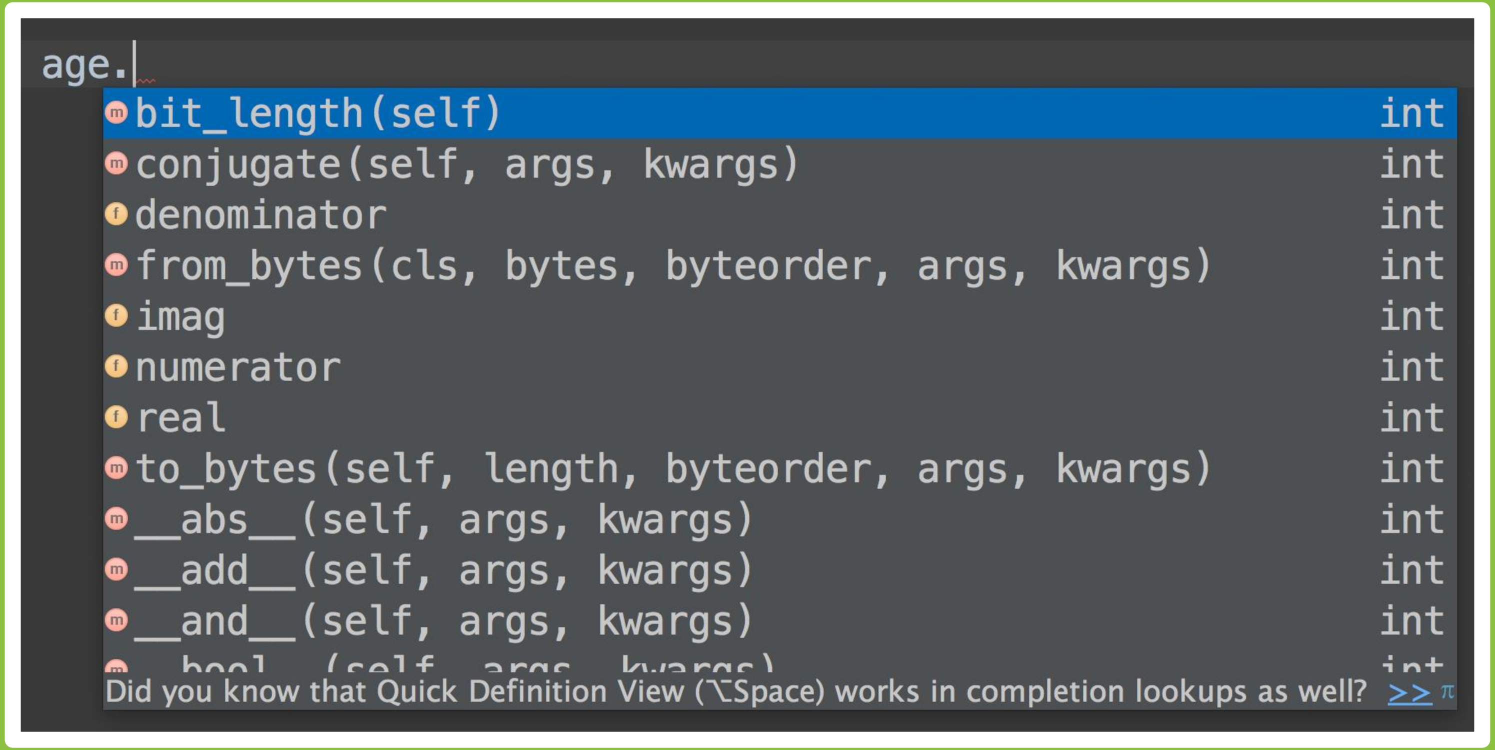


```
age.
```

No suggestions

Python 3

```python
@given("my name is {name:s} and I'm {age:d} years old")
def my_name_and_age(context, name: str, age: int):
```



```
age.
bit_length(self)                                          int
conjugate(self, args, kwargs)                             int
denominator                                               int
from_bytes(cls, bytes, byteorder, args, kwargs)           int
imag                                                      int
numerator                                                 int
real                                                      int
to_bytes(self, length, byteorder, args, kwargs)           int
__abs__(self, args, kwargs)                               int
__add__(self, args, kwargs)                               int
__and__(self, args, kwargs)                               int
__bool__(self, args, kwargs)                              int
Did you know that Quick Definition View (⌥Space) works in completion lookups as well? >> π
```

```python
@given("my name is {name:s} and I'm {age:d} years old")
def my_name_and_age(context, name: str, age: int):
```

```python
@given("my name is {name:s} and I'm {age:d} years old")
def my_name_and_age(context, name: str, age: int):
```

Goat

```python
@given("my name is {name} and I'm {age} years old")
def my_name_and_age(name: str, age: int) -> Person:
    context.person = Person(name, age)
```

Implicit parameters

```python
@then("assert the given person is {} years old")
def assert_person_is_n_years_old(age: int , context: Context):
    assert context.person.age == age
```

```python
@given("my name is {name} and I'm {age} years old")
def my_name_and_age(name: str, age: int):
    context.person = Person(name, age)
```

```python
@given("my name is {name} and I'm {age} years old")
def my_name_and_age(name: str, age: int) -> Person:
    return Person(name, age)
```

```python
@given("my name is {name} and I'm {age} years old")
def my_name_and_age(name: str, age: int) -> Person:
    return Person(name, age)
```

Implicit parameters

```
@then("assert the given person is {} years old")
def assert_person_is_n_years_old(age: int , context: Context):
    assert context.person.age == age
```

Implicit parameters

```
@then("assert the given person is {} years old")
def assert_person_is_n_years_old(age: int , context: Context):
    assert context.person.age == age
```

Implicit parameters

```python
@then("assert the given person is {} years old")
def assert_person_is_n_years_old(age: int , person: Person):
    assert context.person.age == age
```

Implicit parameters

```
@then("assert the given person is {} years old")
def assert_person_is_n_years_old(age: int , person: Person):
    assert context.person.age == age
```

Implicit parameters

```python
@then("assert the given person is {} years old")
def assert_person_is_n_years_old(age: int , person: Person):
    assert person.age == age
```

Implicit parameters

```
@then("assert the given person is {} years old")
def assert_person_is_n_years_old(age: int , person: Person):
    assert person.age == age
```

```
pip install goat
```

github.com/cuescience/goat

@EljeyRedHair