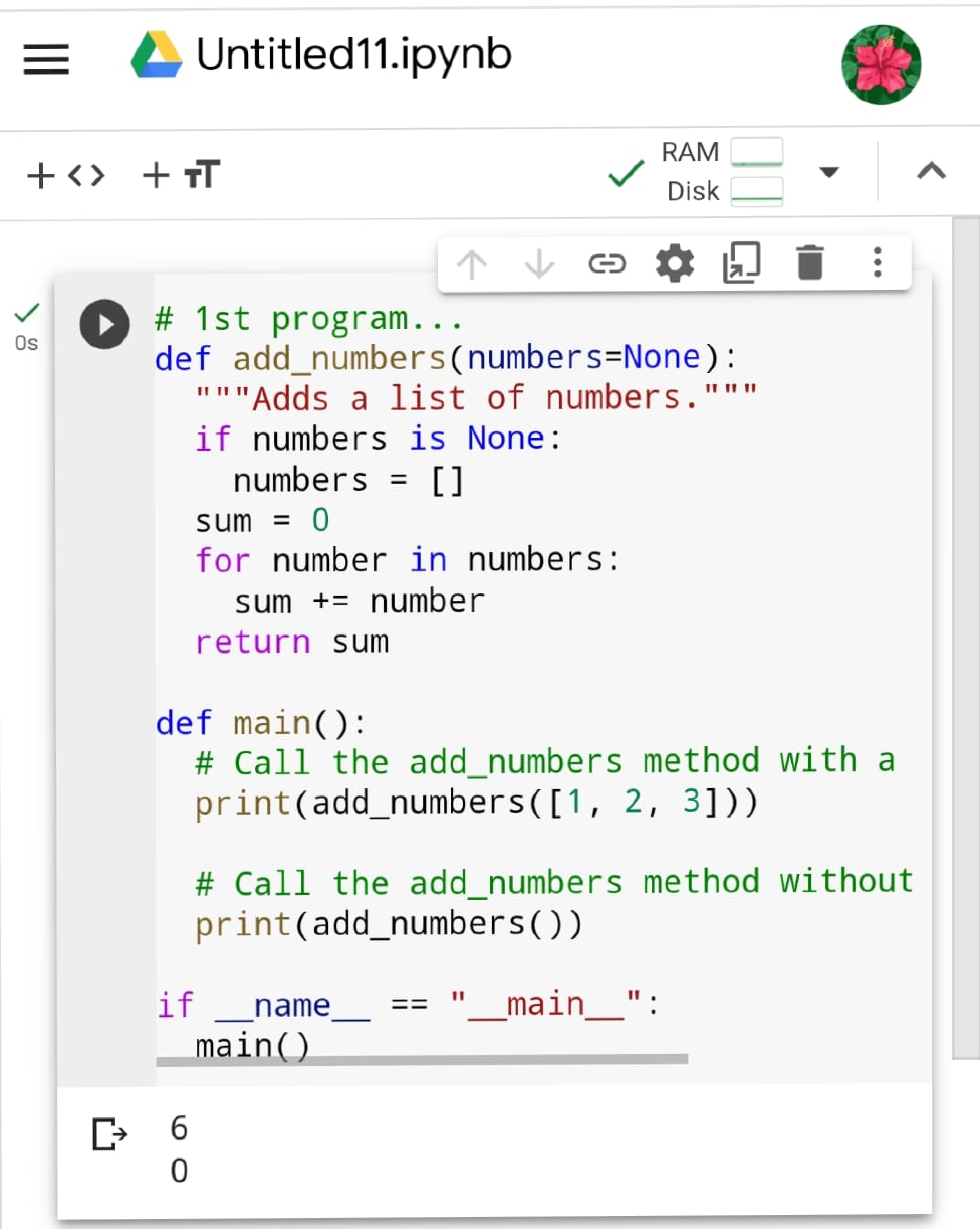
13.Method Overloading

1. Write two methods with the same name but different number of parameters of same type and call the methods

2. Write two methods with the same name but different number of parameters of different data type and call the methods

3. Write two methods with the same name and same number of parameters of same type



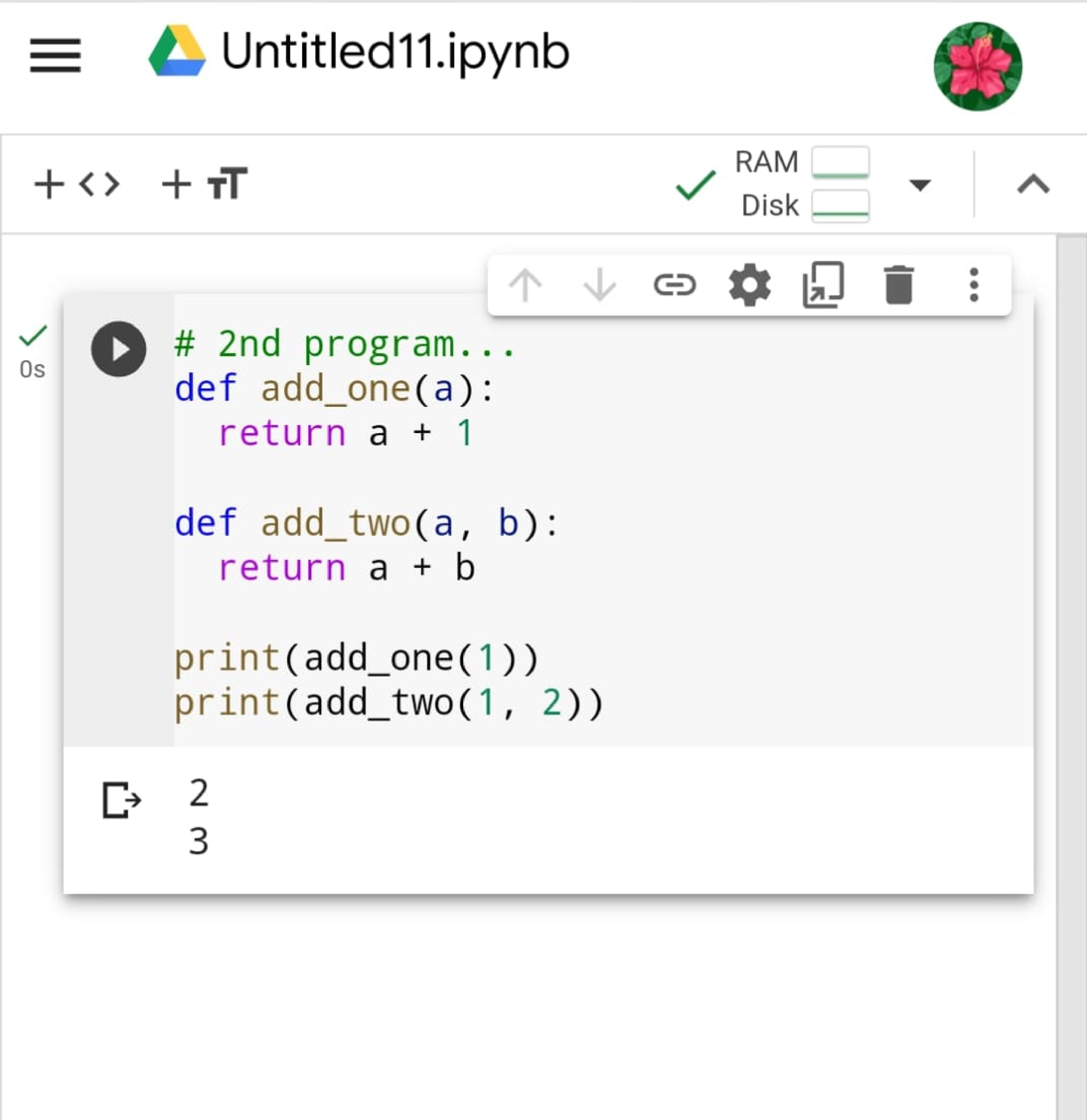
```python
# 1st program...
def add_numbers(numbers=None):
  """Adds a list of numbers."""
  if numbers is None:
    numbers = []
  sum = 0
  for number in numbers:
    sum += number
  return sum

def main():
  # Call the add_numbers method with a
  print(add_numbers([1, 2, 3]))

  # Call the add_numbers method without
  print(add_numbers())

if __name__ == "__main__":
  main()
```

```
6
0
```

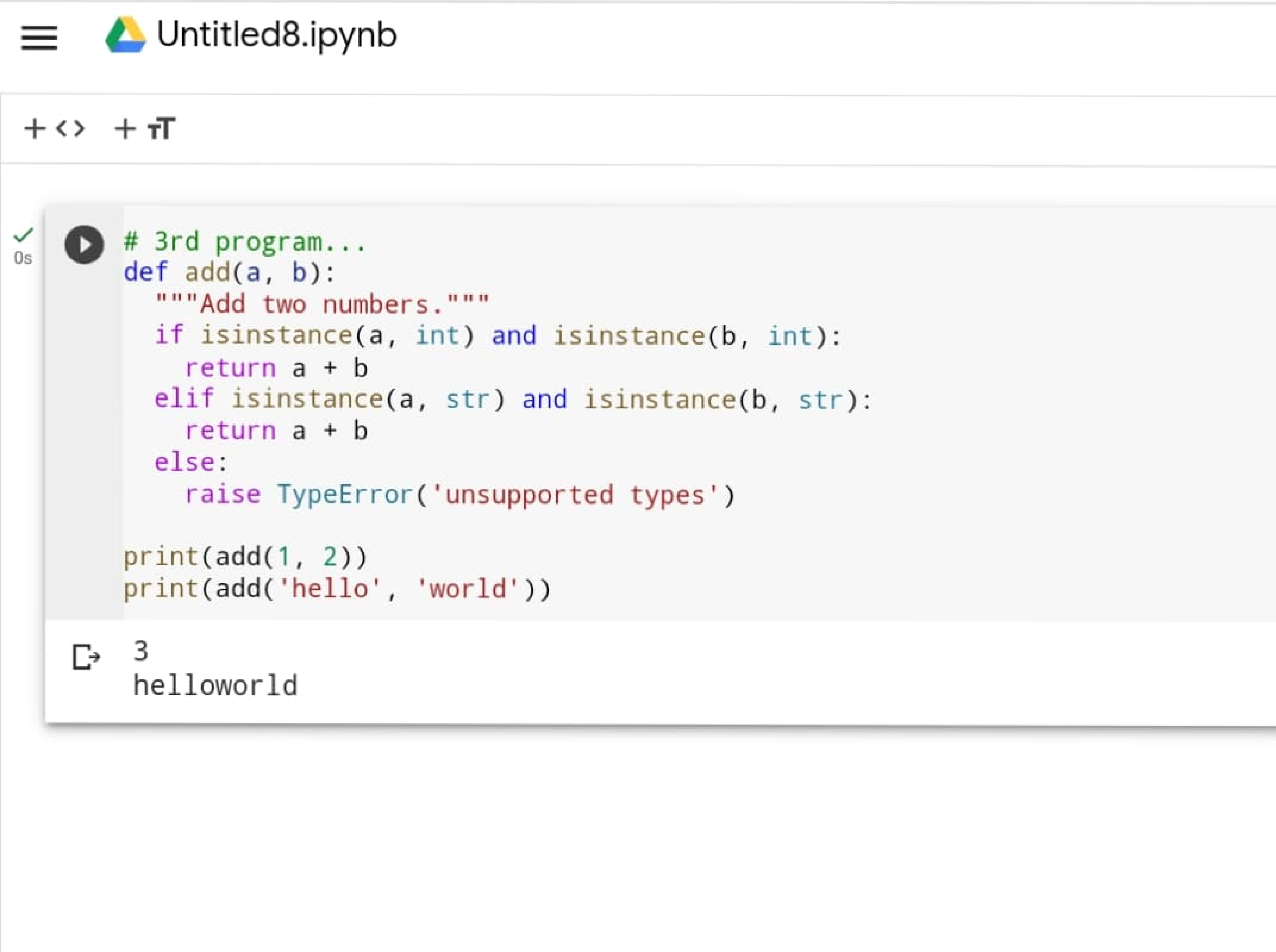


```python
# 2nd program...
def add_one(a):
  return a + 1

def add_two(a, b):
  return a + b

print(add_one(1))
print(add_two(1, 2))
```

```
2
3
```



```python
# 3rd program...
def add(a, b):
  """Add two numbers."""
  if isinstance(a, int) and isinstance(b, int):
    return a + b
  elif isinstance(a, str) and isinstance(b, str):
    return a + b
  else:
    raise TypeError('unsupported types')

print(add(1, 2))
print(add('hello', 'world'))
```

```
3
helloworld
```