

Ast tree after dumping it:

```
ast tree:
Module(
  body=[
    FunctionDef(
      name='AddNumbers',
      args=arguments(
        posonlyargs=[],
        args=[
          arg(arg='a'),
          arg(arg='b')],
        kwonlyargs=[],
        kw_defaults=[],
        defaults=[]),
      body=[
        Return(
          value=BinOp(
            left=Name(id='a', ctx=Load()),
            op=Add(),
            right=Name(id='b', ctx=Load()))),
        decorator_list=[])],
      type_ignores=[])
```

Breakdown / Annotation:

New Body of a class

ast tree:

Module(

body=[

FunctionDef(

name='AddNumbers',

args=arguments(

posonlyargs=[],

args=[

arg(arg='a'),

arg(arg='b')],

kwonlyargs=[],

kw_defaults=[],

defaults=[]),

body=[

Return(

value=BinOp(

left=Name(id='a', ctx=Load()),

op=Add(),

right=Name(id='b', ctx=Load()))],

decorator_list=[]),

type_ignores=[])

Python module where
Function is stored in

Function Name

Function Definition: where the
function is declared / defined

Breakdown / Annotation:

ast tree:

```
Module(  
  body=[  
    FunctionDef(  
      name='AddNumbers',  
      args=arguments(  
        posonlyargs=[],  
        args=[  
          arg(arg='a'),  
          arg(arg='b')],  
        kwonlyargs=[],  
        kw_defaults=[],  
        defaults=[]),  
      body=[  
        Return(  
          value=BinOp(  
            left=Name(id='a', ctx=Load()),  
            op=Add(),  
            right=Name(id='b', ctx=Load()))],  
        decorator_list=[]),  
      type_ignores=[])
```

children of
args

Arguments / Parameters class
The space where all args
are stored

Breakdown / Annotation:

ast tree:

```
Module(
```

```
  body=[
```

```
    FunctionDef(
```

```
      name='AddNumbers',
```

```
      args=arguments(
```

```
        posonlyargs=[],
```

```
        args=[
```

```
          arg(arg='a'),
```

```
          arg(arg='b')],
```

```
        kwonlyargs=[],
```

```
        kw_defaults=[],
```

```
        defaults=[]),
```

```
      body=[
```

```
        Return(
```

```
          value=BinOp(
```

```
            left=Name(id='a', ctx=Load()),
```

```
            op=Add(),
```

```
            right=Name(id='b', ctx=Load()))),
```

```
        decorator_list=[])],
```

```
      type_ignores=[])
```

return a+b

Function Body:

where the function stores code
of the function

Visual of tree (without the extreme natures)

