

```
def parent():  
    chaild1 = 10  
    print(chaild1)  
    return chaild1
```

```
result = parent()  
print(result)
```

Parent ↘

```
chaild1 = 10  
print(chaild1)  
return chaild1
```

```
def parent():  
    chaid1 = 10  
    print(chaid1)  
    return chaid1
```

```
result = parent()  
print(result)
```

10 140715073346128

10 140715073346128

Parent →

```
chaid1 = 10  
print(chaid1)  
return chaid1
```

~~chaid1~~ →

result →

10

6128

Parent function definition

```
def parent():
```

```
    def chaild():
```

```
        print('chaild function definition')
```

```
    chaild()
```

→ Parent function

Block of Code

Parent function call

```
def parent():
```

```
    def chaild():
```

```
        print('chaild function definition')
```

```
    chaild()
```

→ Parent function

Block of Code

Parent ()

→ chaild function call

→ chaild function Block of Code

Parent definition

```
def parent():  
    print('parent function definition start')  
    def chaild():  
        print('chaild function definition')  
    print('parent function definition end')  
    return chaild  
  
fun = parent()
```

```
parent function definition start  
parent function definition end
```

```
fun()
```

```
chaild function definition
```

Parent ↘

```
print('parent function definition start')  
def chaild():  
    print('chaild function definition')  
print('parent function definition end')  
return chaild
```

Parent call

```
def parent():  
    print('parent function definition start')  
    def chaild():  
        print('chaild function definition')  
    print('parent function definition end')  
    return chaild
```

```
fun = parent()
```

```
parent function definition start ✓  
parent function definition end ✓
```

```
fun()
```

```
chaild function definition
```

Parent →

```
1 print('parent function definition start')  
2 def chaild():  
    print('chaild function definition')  
3 print('parent function definition end')  
4 return chaild
```

chaild function definition.

chaild →

```
print('chaild function definition')
```

Parent call

```
def parent():  
    print('parent function definition start')  
    def chaild():  
        print('chaild function definition')  
    print('parent function definition end')  
    return chaild
```

chaild
fun = parent()

parent function definition start ✓
parent function definition end ✓

fun()

chaild function definition

Parent →

```
1 print('parent function definition start')  
2 def chaild():  
    print('chaild function definition')  
3 print('parent function definition end')  
4 return chaild
```

chaild function definition.

~~chaild~~ →
fun →

```
print('chaild function definition')
```


Parent call

```
def parent():  
    print('parent function definition start')  
    def chaild():  
        print('chaild function definition')  
    print('parent function definition end')  
    return chaild
```

chaild
fun = parent()

parent function definition start ✓
parent function definition end ✓

fun()

chaild function definition

Parent →

```
1 print('parent function definition start')  
2 def chaild():  
    print('chaild function definition')  
3 print('parent function definition end')  
4 return chaild
```

chaild function definition.

~~chaild~~

fun →

```
print('chaild function definition')
```

chaild call


```
def testcode(i):  
    def process():  
        print(i)  
    return process
```

testcode ↘

```
def process():  
    print(i)  
return process
```

```
def testcode(i):  
    def process():  
        print(i)  
    return process
```

```
showvalue1 = testcode(10)
```

testcode(i=10)

```
def process():  
    print(i)  
    return process
```

- ① process function will be defined
- ② returns process function name

~~process~~

Print(10)

← showvalue1

```
def testcode(i):  
    def process():  
        print(i)  
    return process
```

```
showvalue1 = testcode(10)
```

```
showvalue2 = testcode(20)
```

~~testcode(i=20)~~

```
def process():  
    print(i)  
    return process
```

- ① process function will be defined
- ② returns process function name

~~process~~ →
Print(10) ← showvalue1

~~process~~ →
Print(20) ← showvalue2

```
def testcode(i):  
    def process():  
        print(i)  
    return process
```

~~testcode(i=Hello)~~

```
def process():  
    print(i)  
    return process
```

- ① process function will be defined
- ② returns process function name

~~process~~ → **Print(10)** ← showvalue1

~~process~~ → **Print(20)** ← showvalue2

~~process~~ → **Print('Hello')** ← showvalue3

showvalue1 = testcode(10)

showvalue2 = testcode(20)

showvalue3 = testcode('hello')