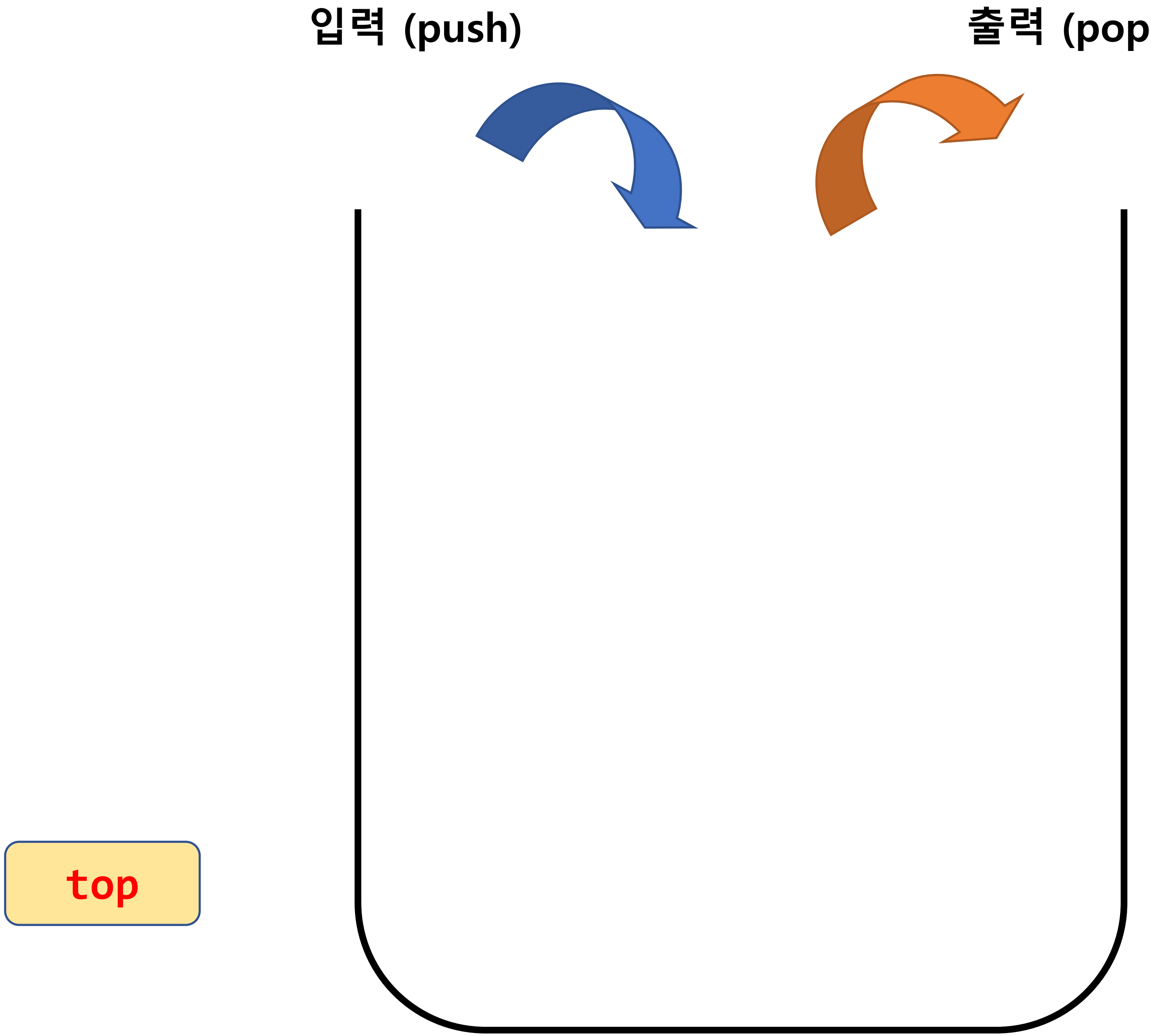
Data Struct

Stack

Stack 구조 파악하기

Data Struct

Stack



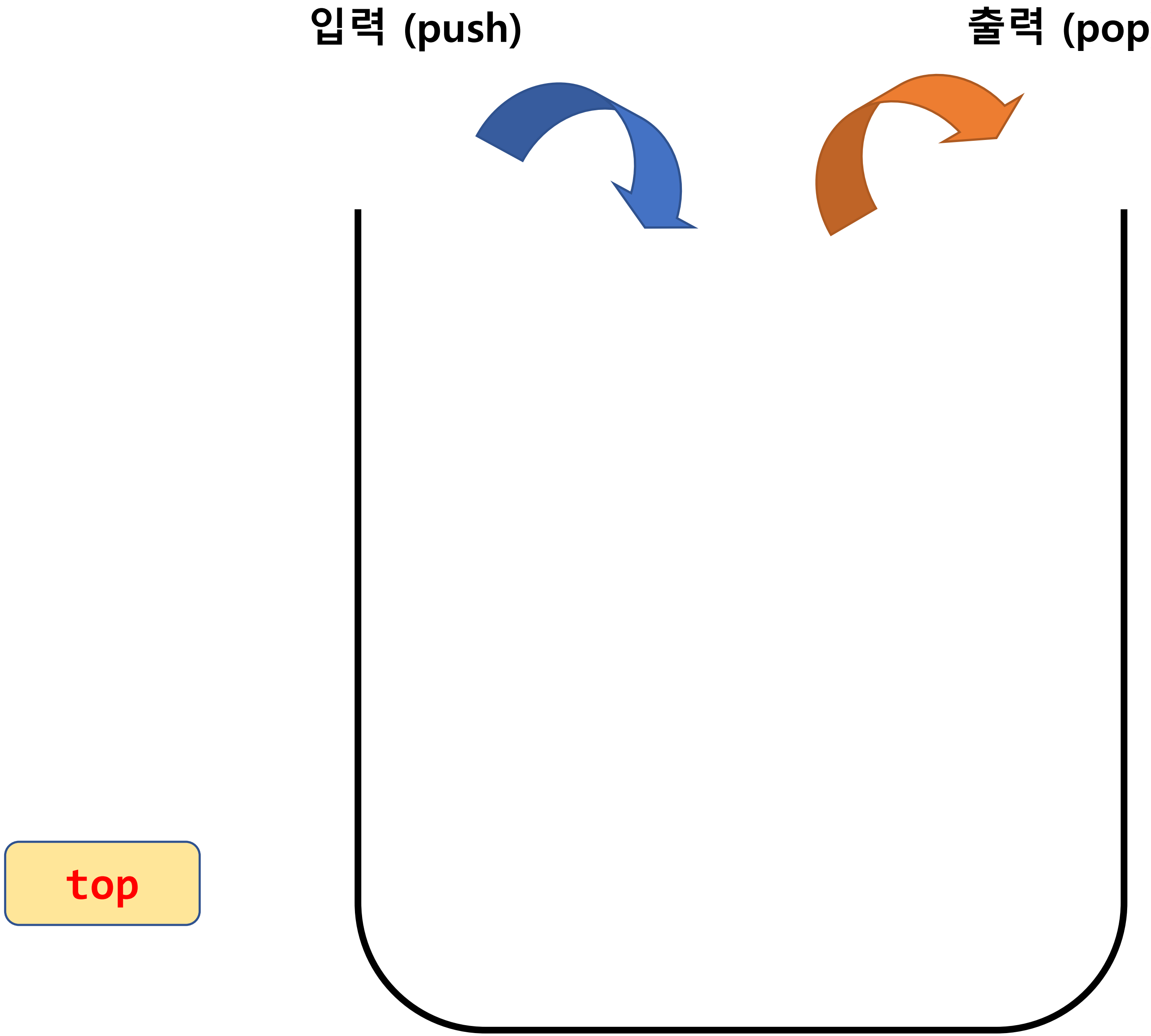
입력 (push)

출력 (pop)

top

Data Struct

Stack



입력 (push)

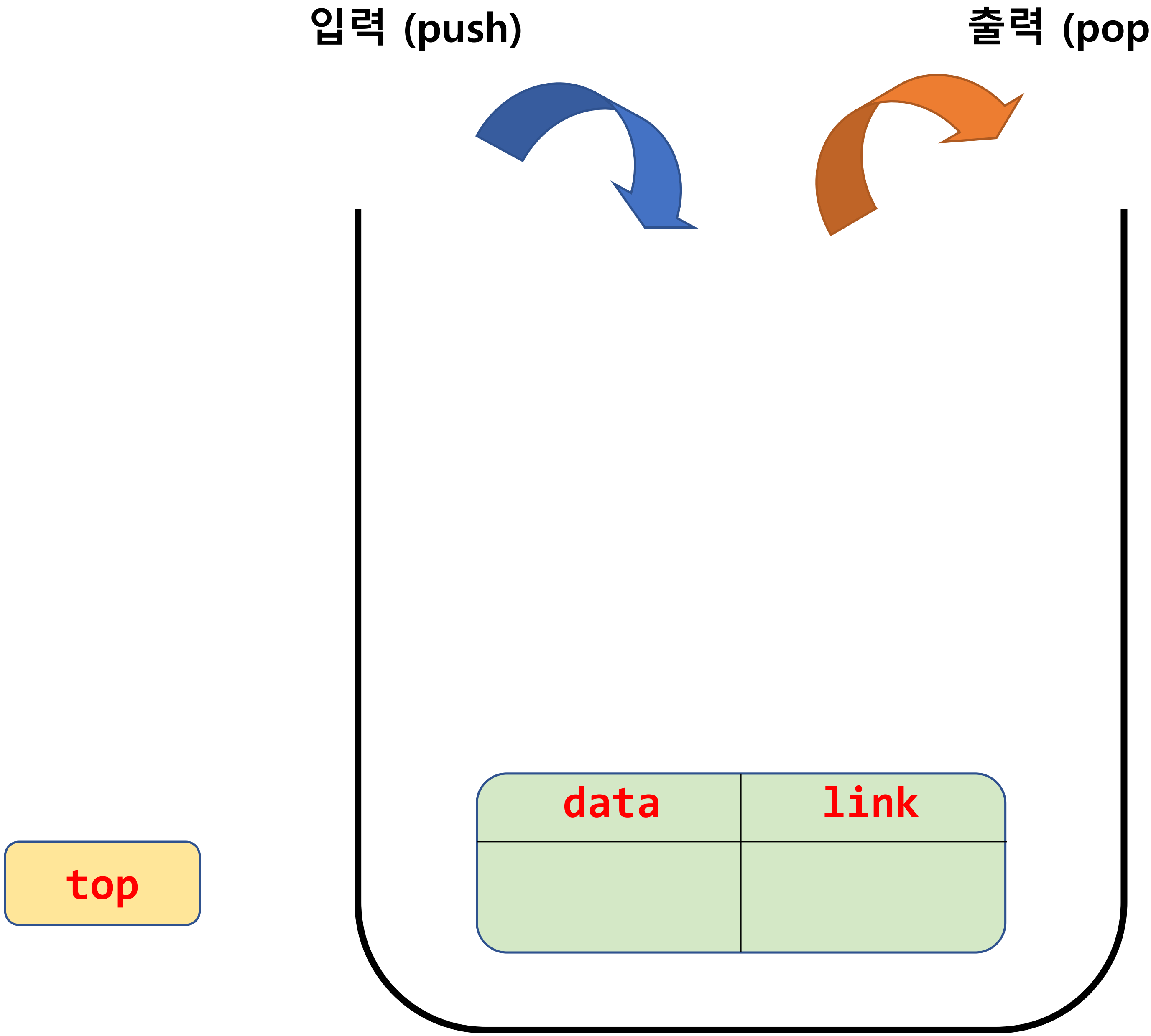
출력 (pop)

top

push(data1);

Data Struct

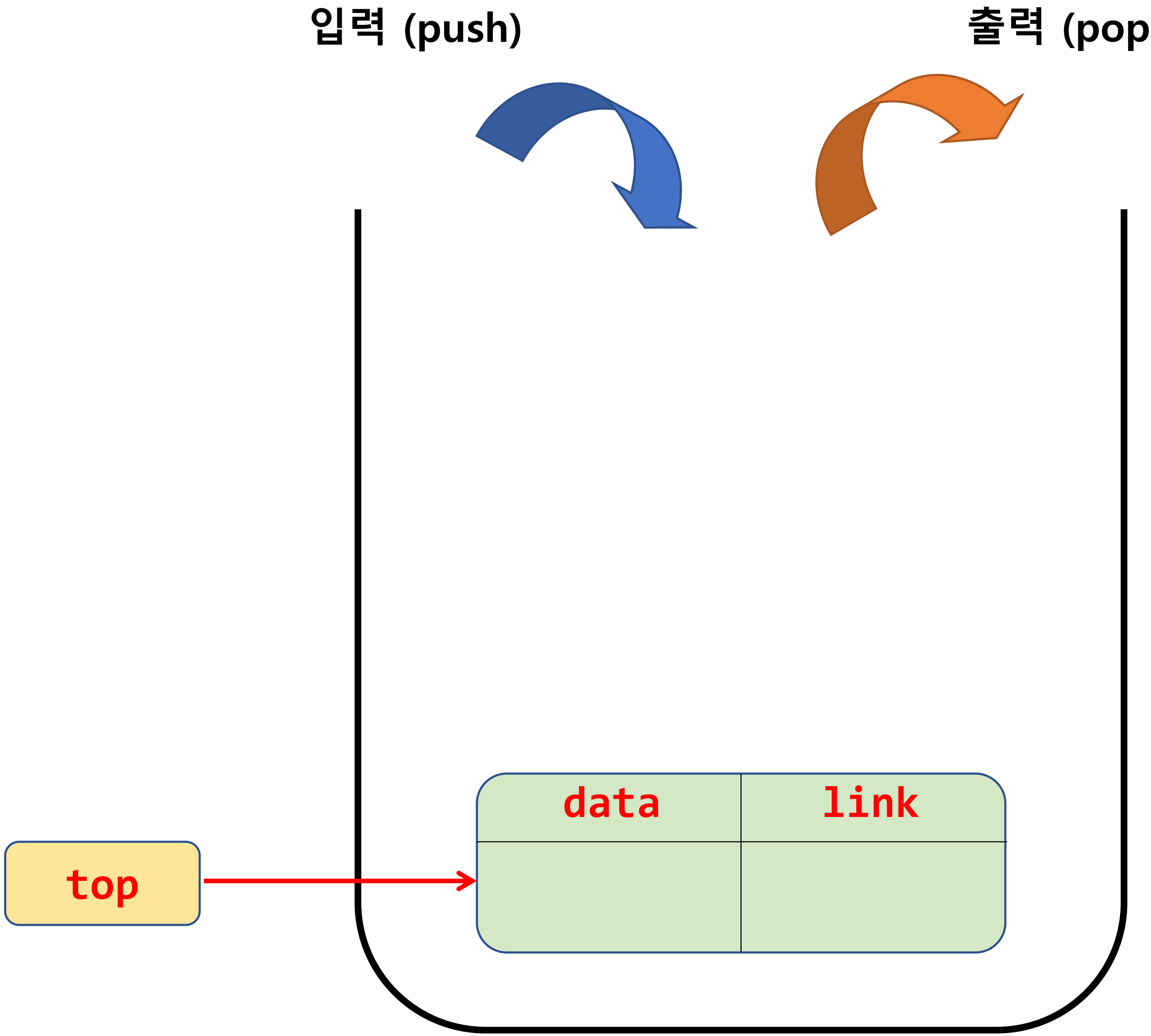
Stack



push(data1);

Data Struct

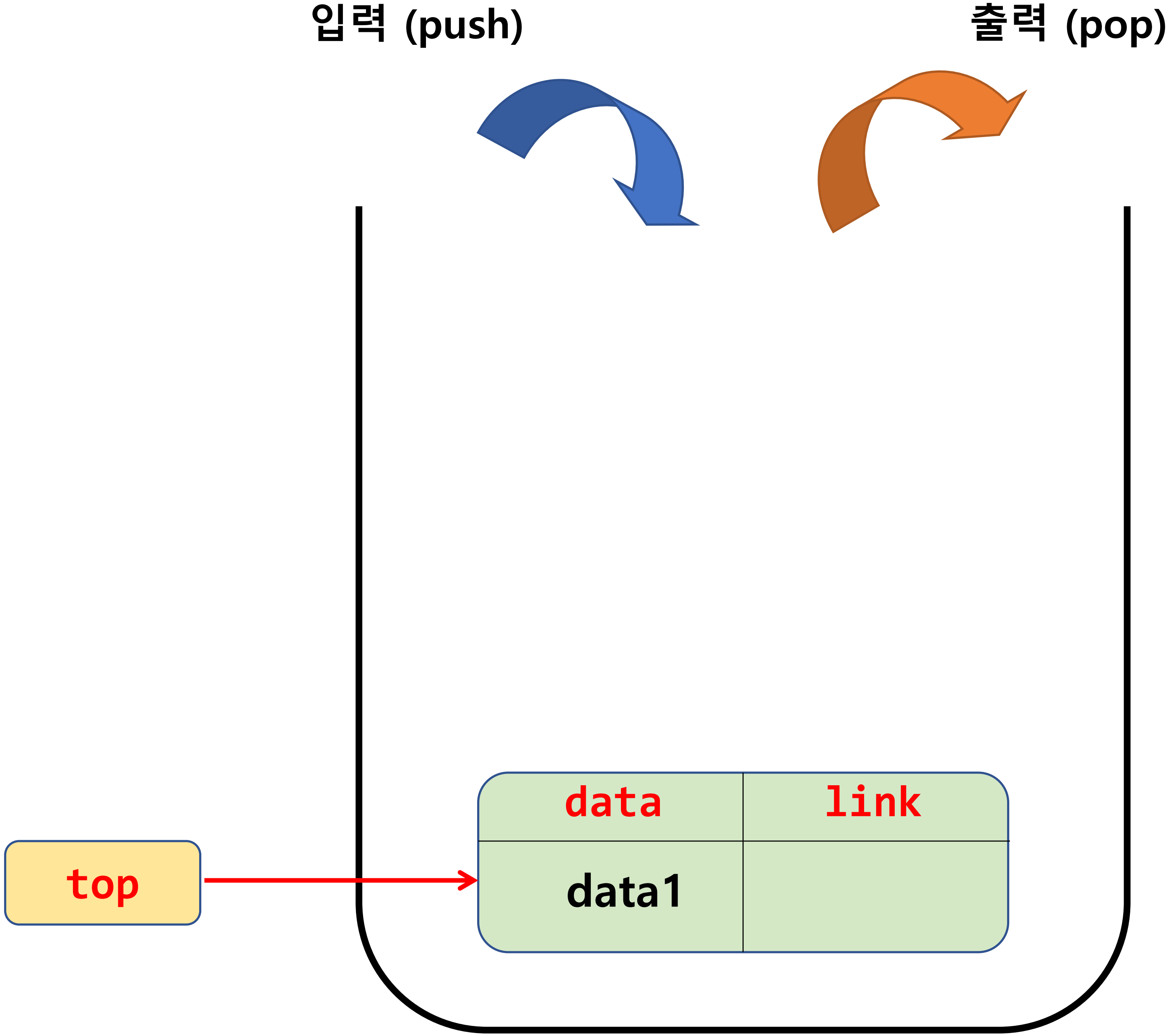
Stack



push(data1);

Data Struct

Stack



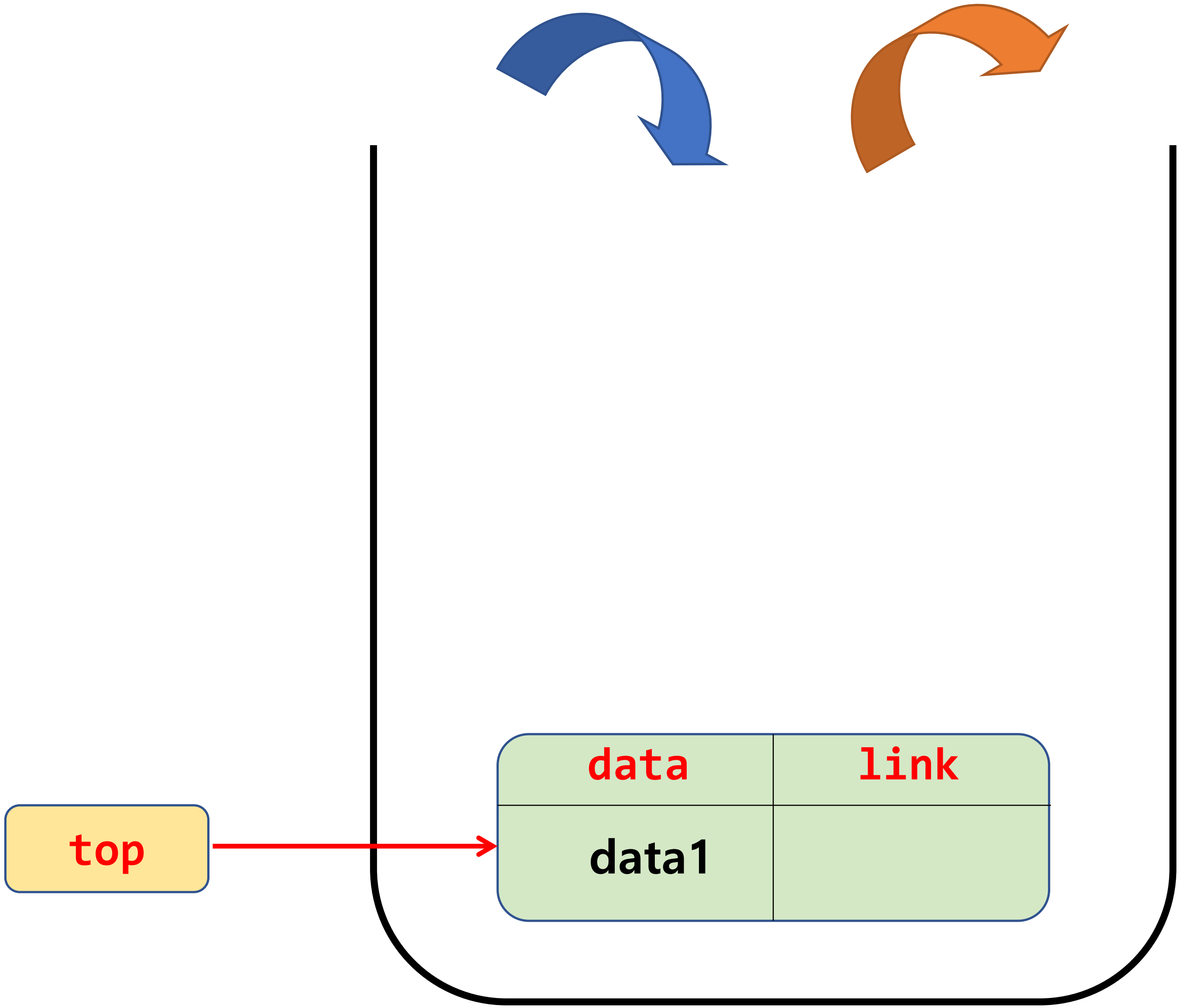
push(data1);

Data Struct

Stack

입력 (push)

출력 (pop)



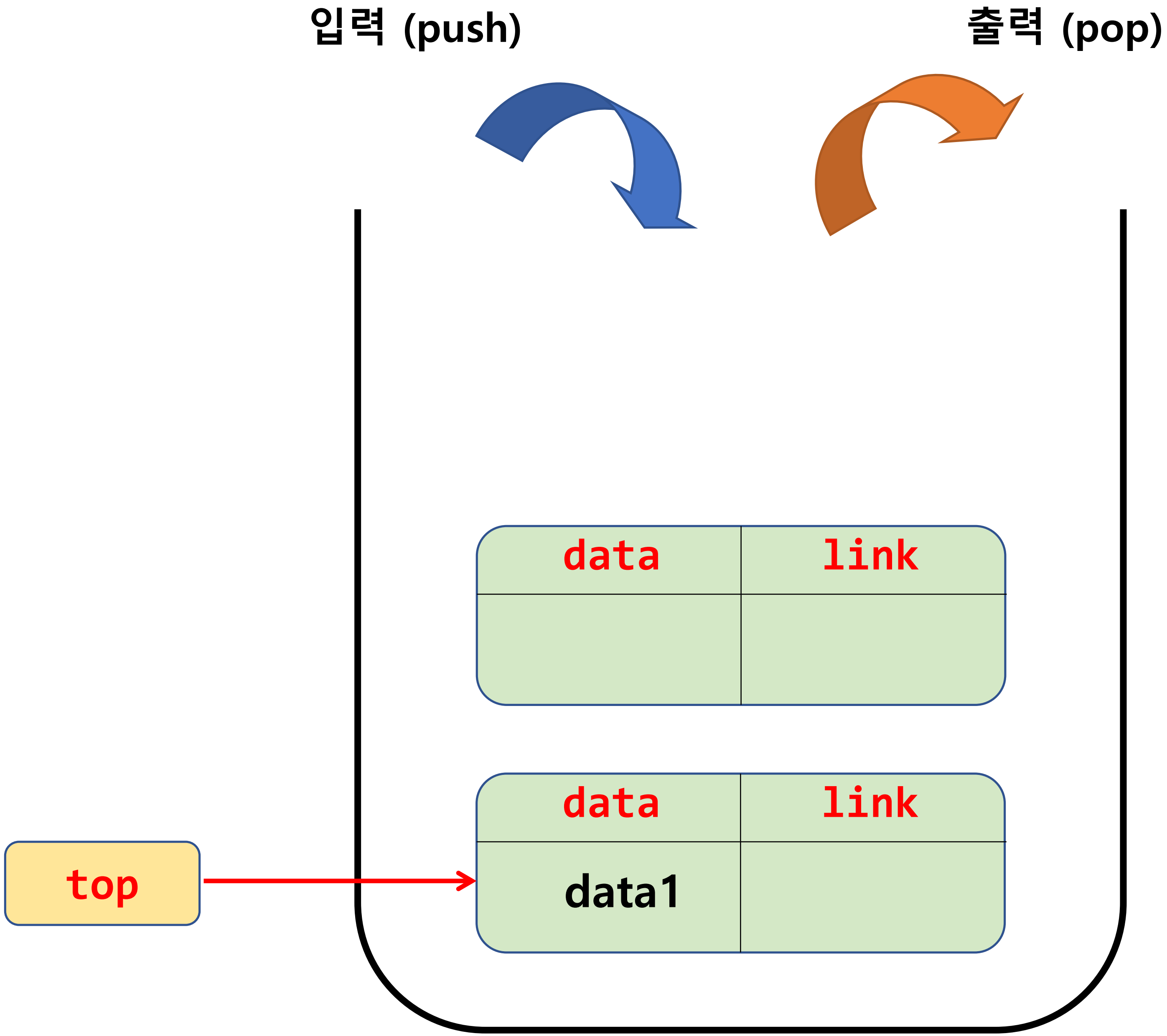
top

data	link
data1	

```
push(data1);

push(data2);
```

Data Struct

Stack

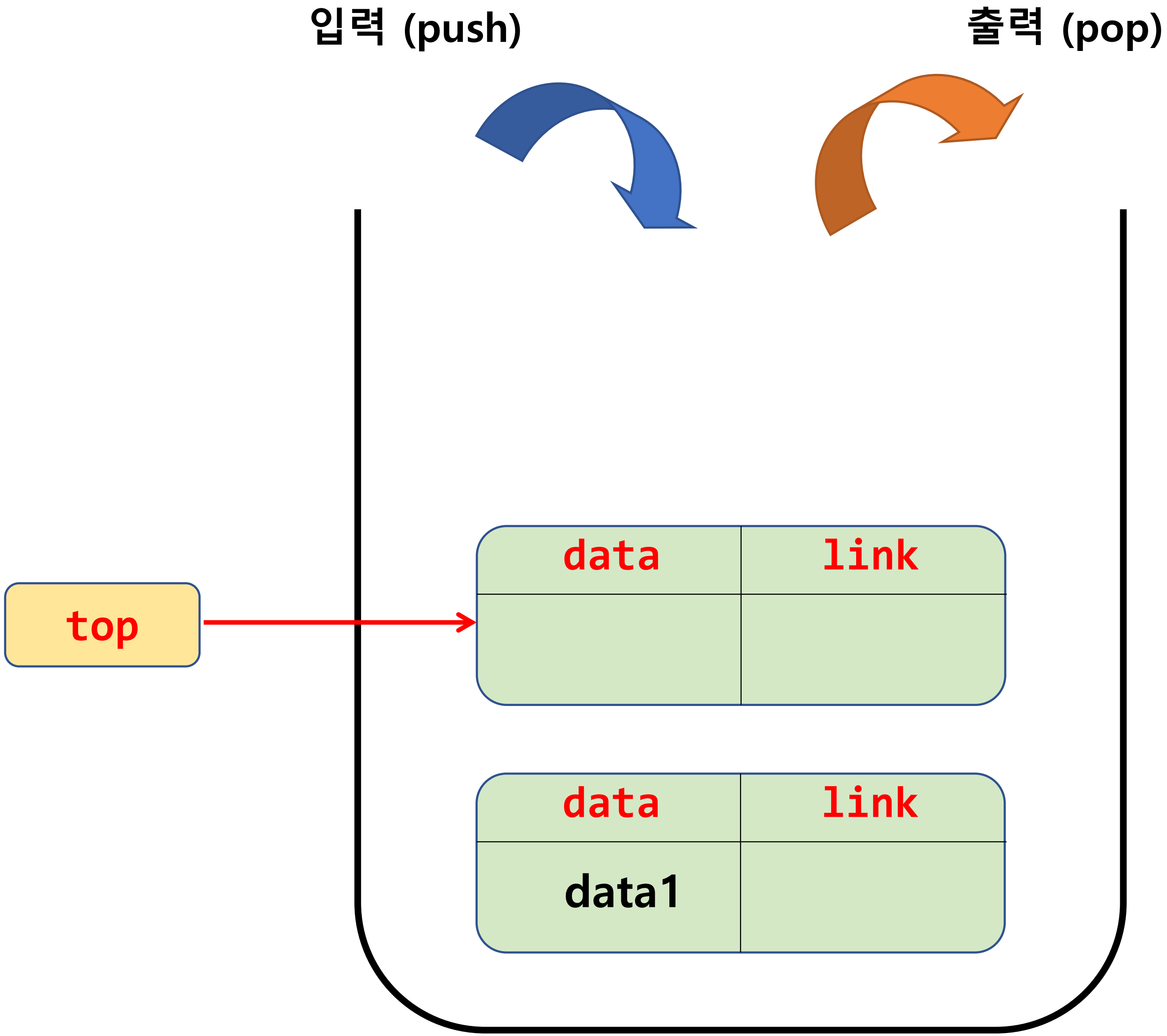


push(data1);

push(data2);

Data Struct

Stack

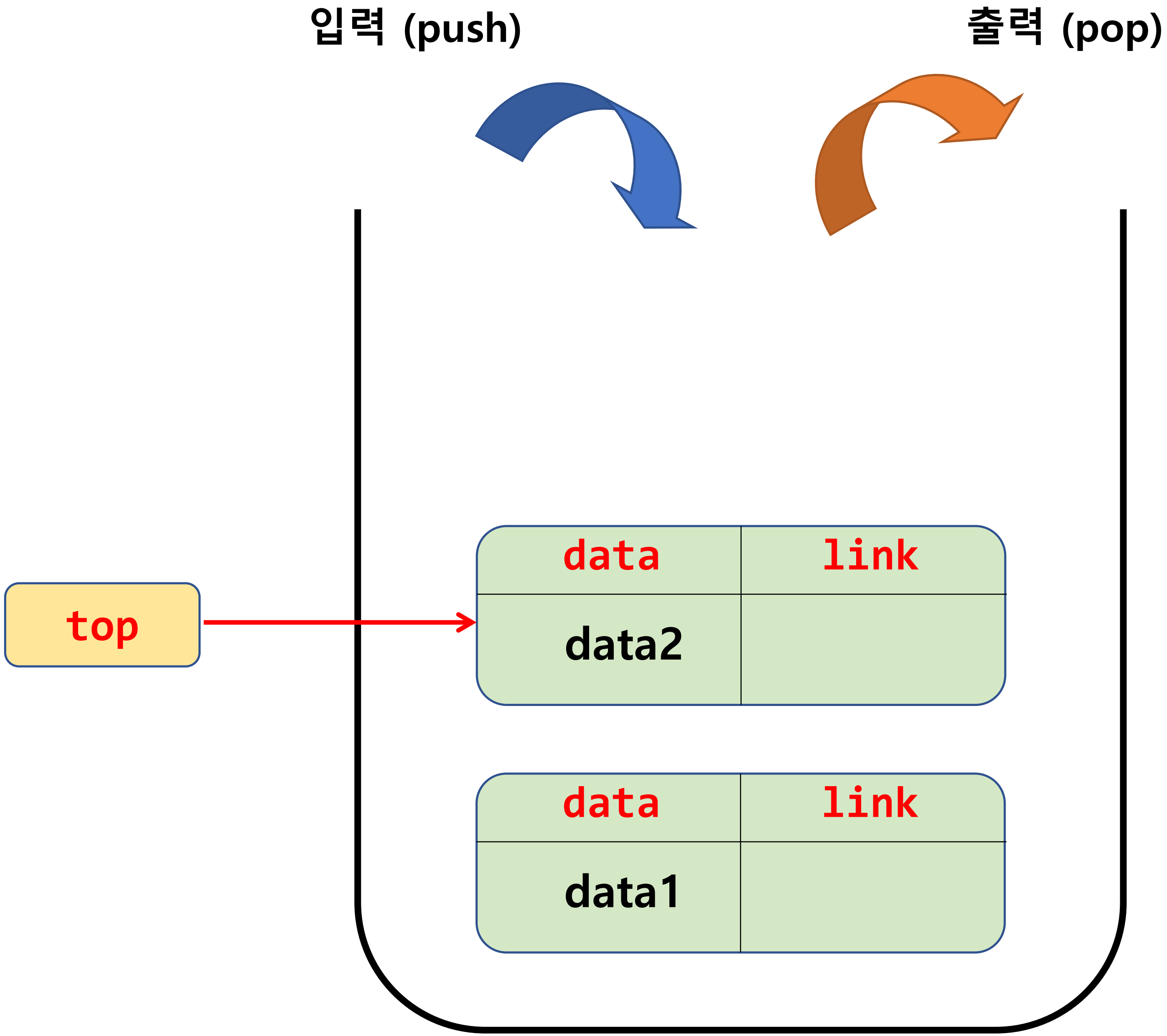


push(data1);

push(data2);

Data Struct

Stack

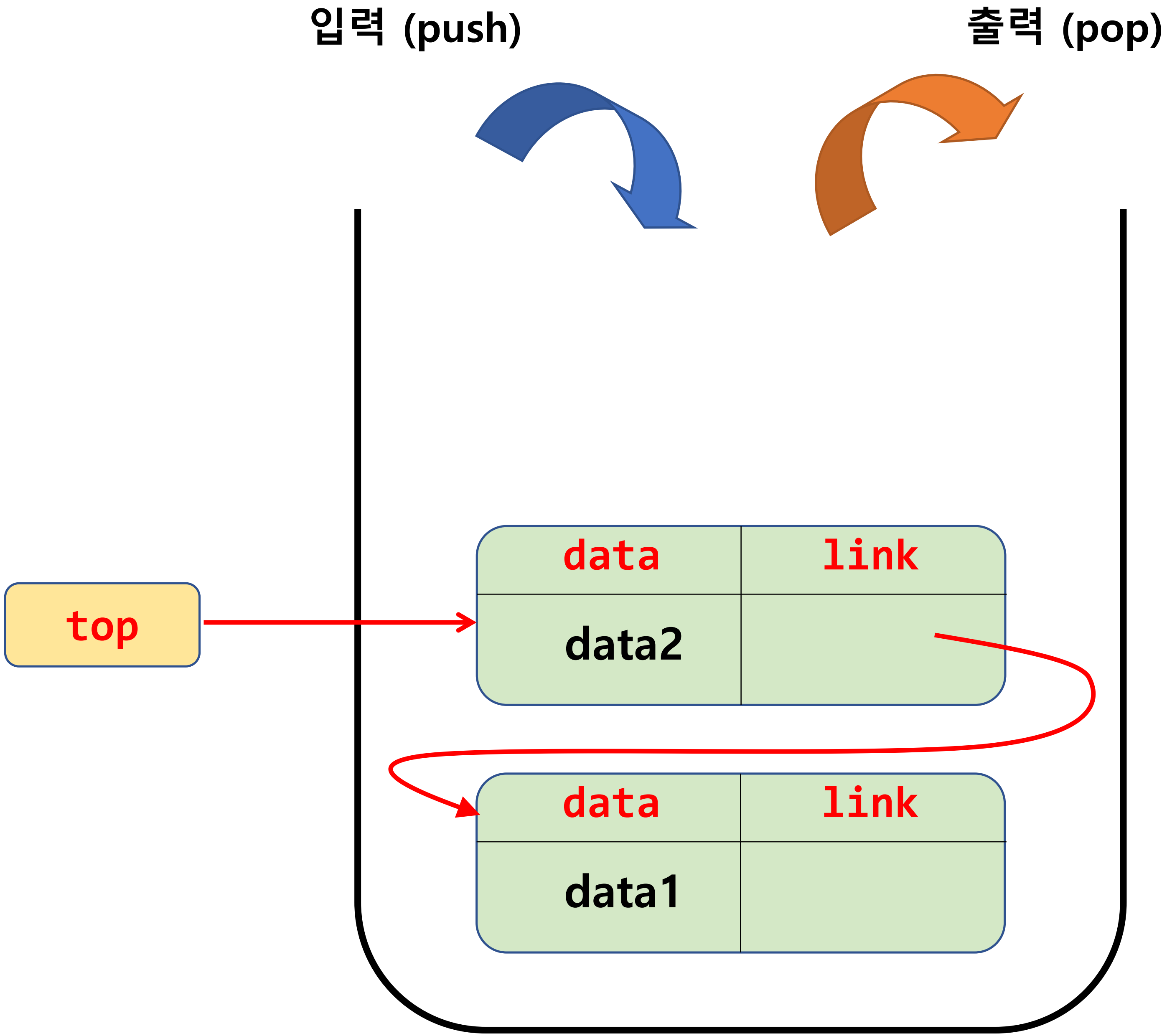


push(data1);

push(data2);

Data Struct

Stack

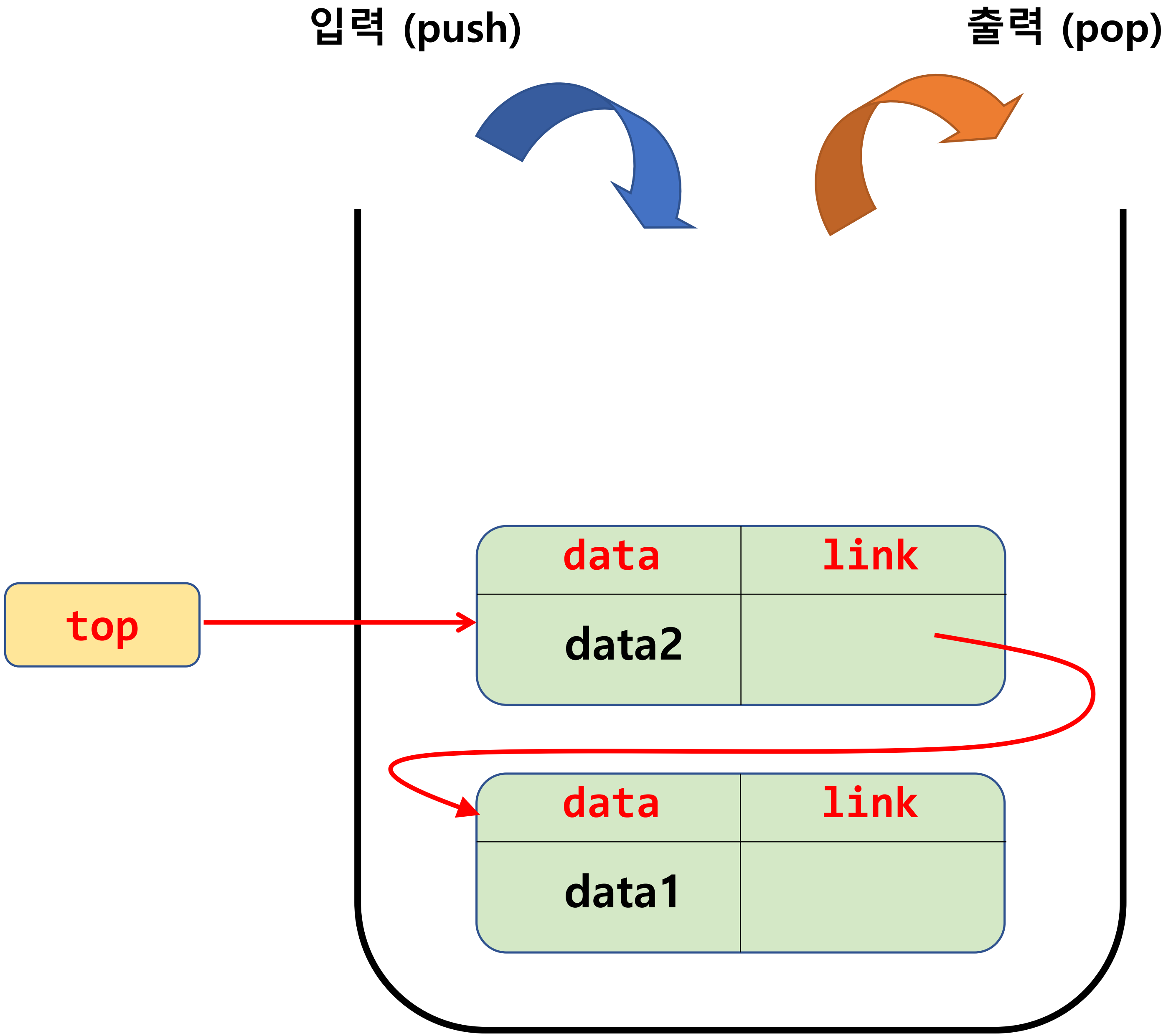


push(data1);

push(data2);

Data Struct

Stack



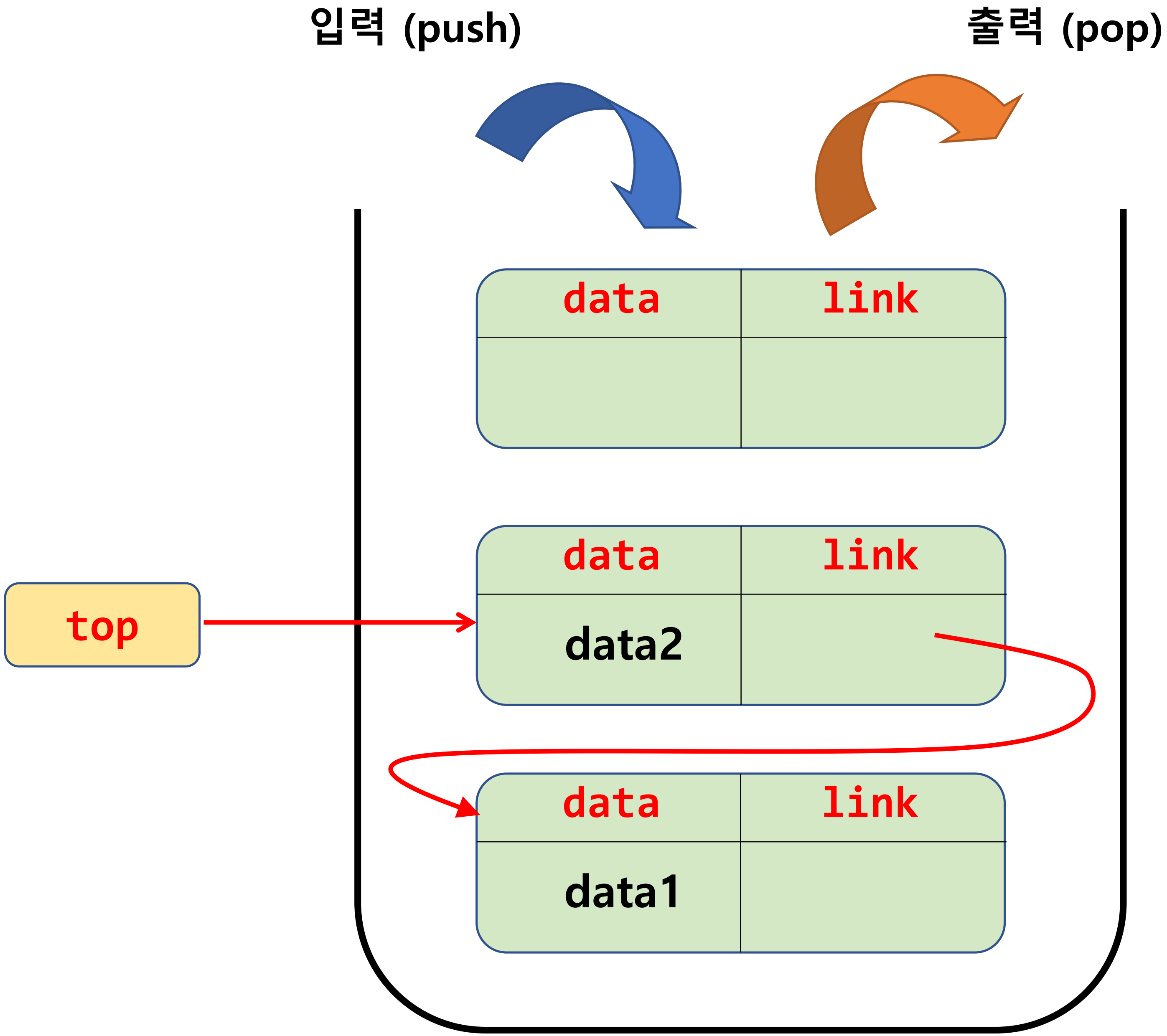
push(data1);

push(data2);

push(data3);

Data Struct

Stack



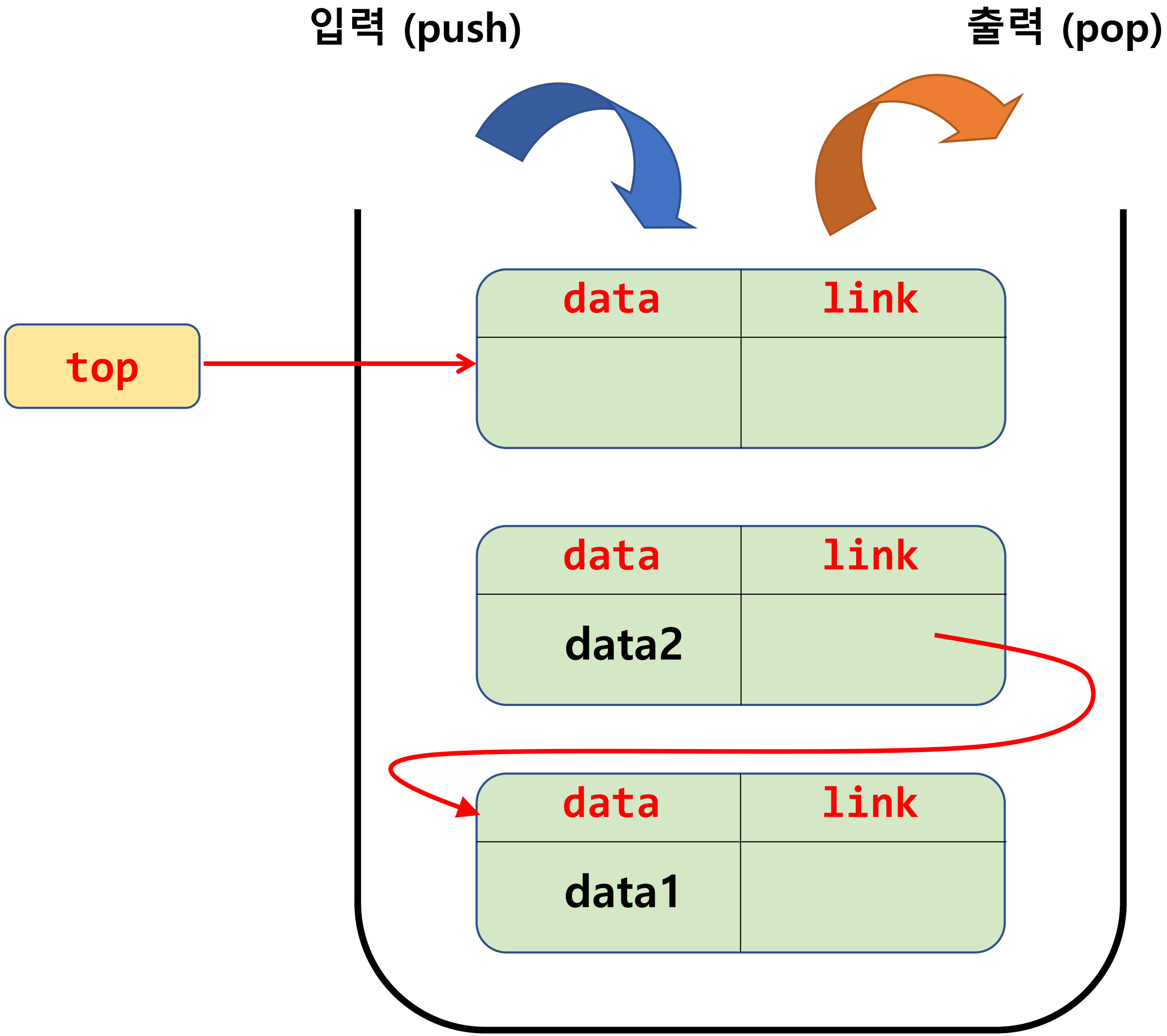
push(data1);

push(data2);

push(data3);

Data Struct

Stack



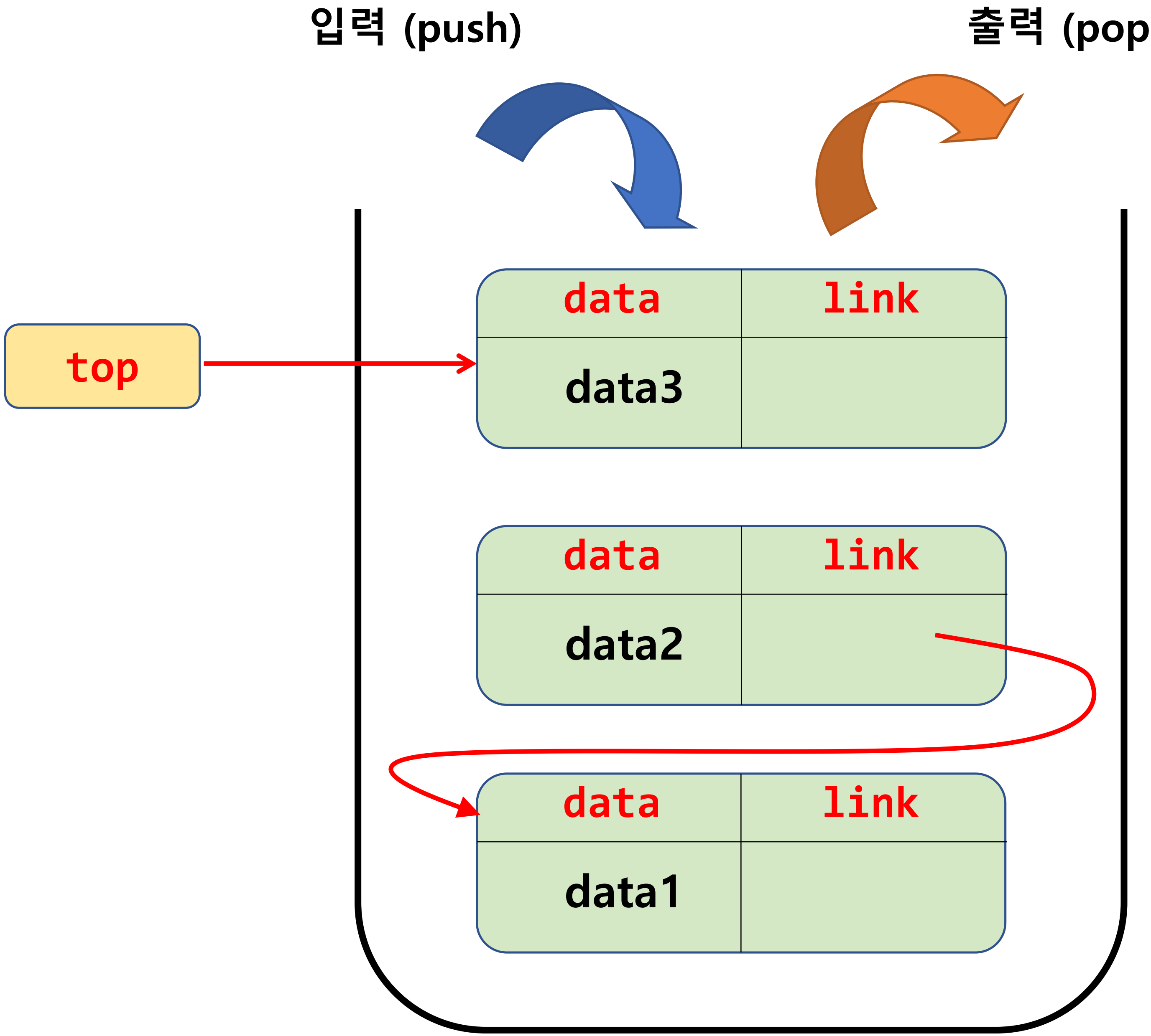
push(data1);

push(data2);

push(data3);

Data Struct

Stack



입력 (push)

출력 (pop)

top

data	link
data3	

data	link
data2	

data	link
data1	

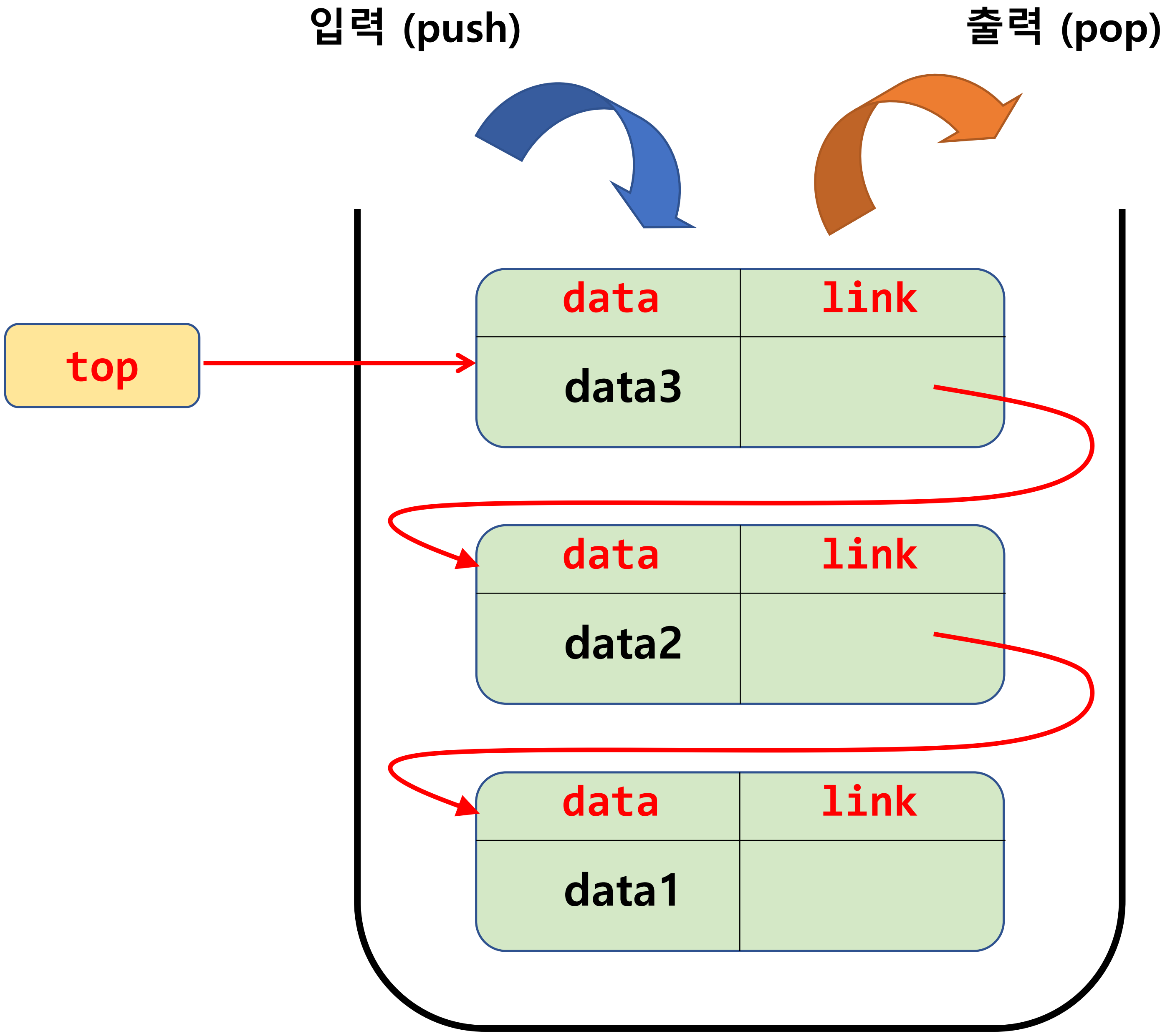
push(data1);

push(data2);

push(data3);

Data Struct

Stack



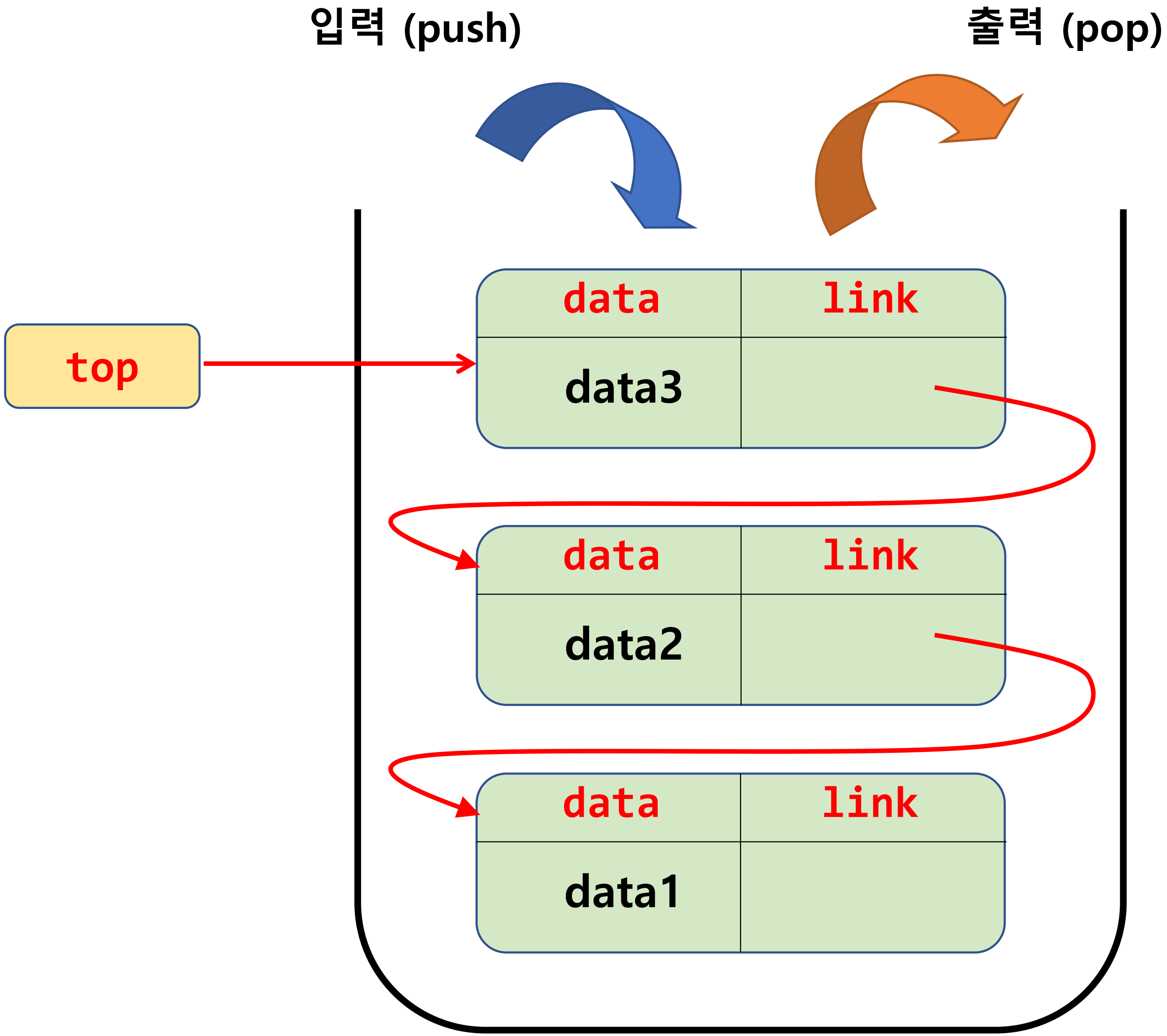
push(data1);

push(data2);

push(data3);

Data Struct

Stack



push(data1);

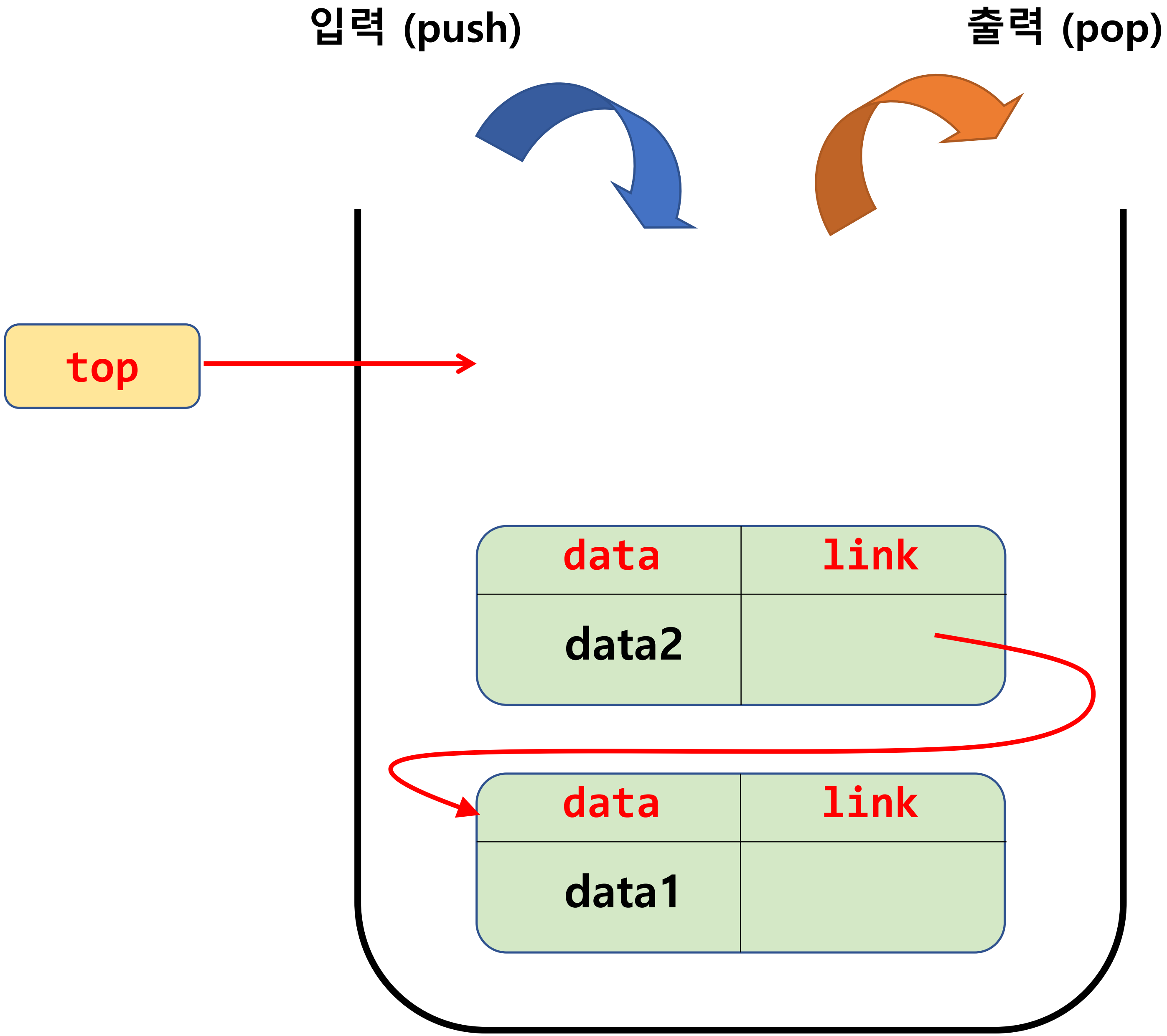
push(data2);

push(data3);

pop();

Data Struct

Stack



push(data1);

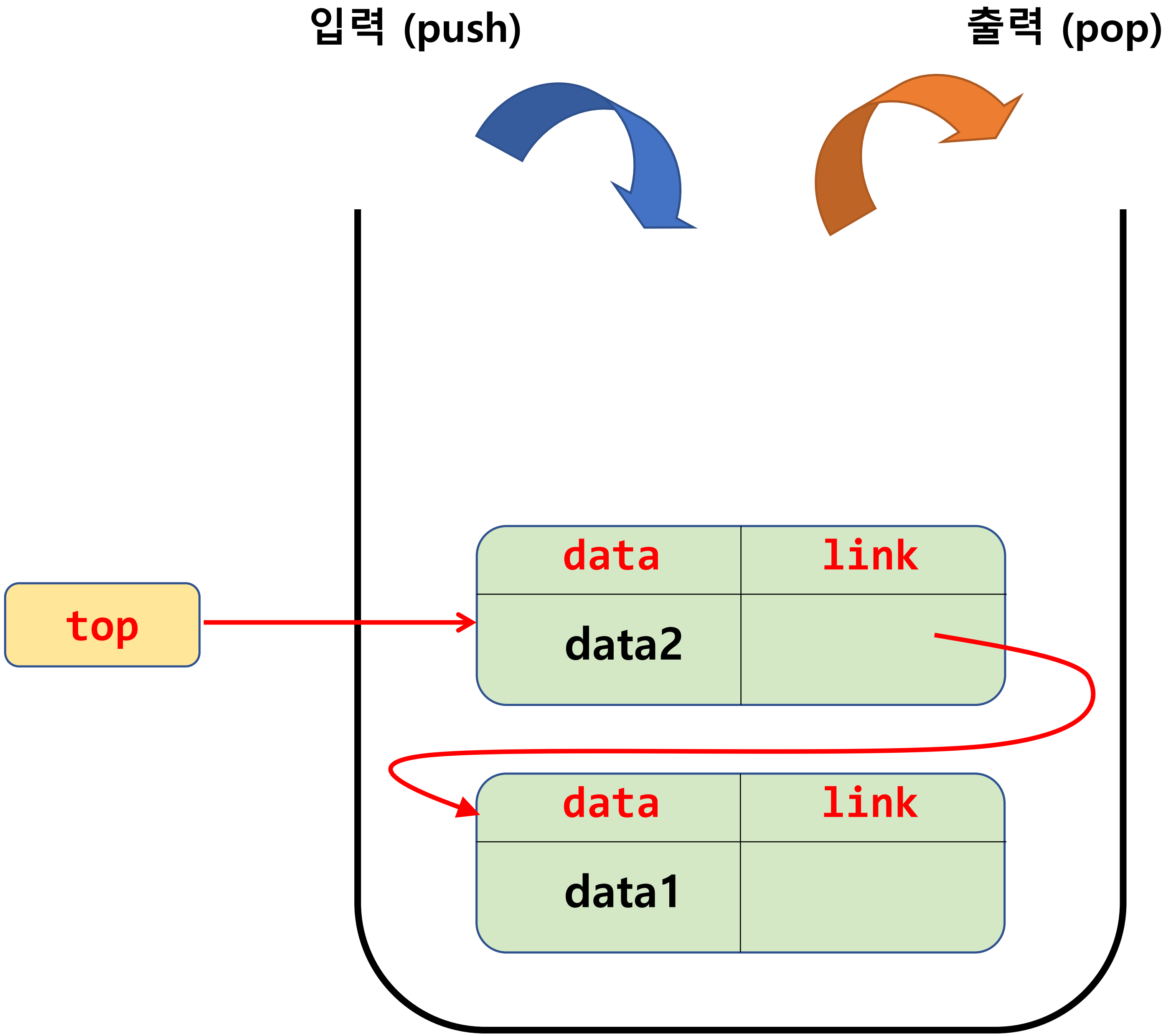
push(data2);

push(data3);

pop();

Data Struct

Stack



push(data1);

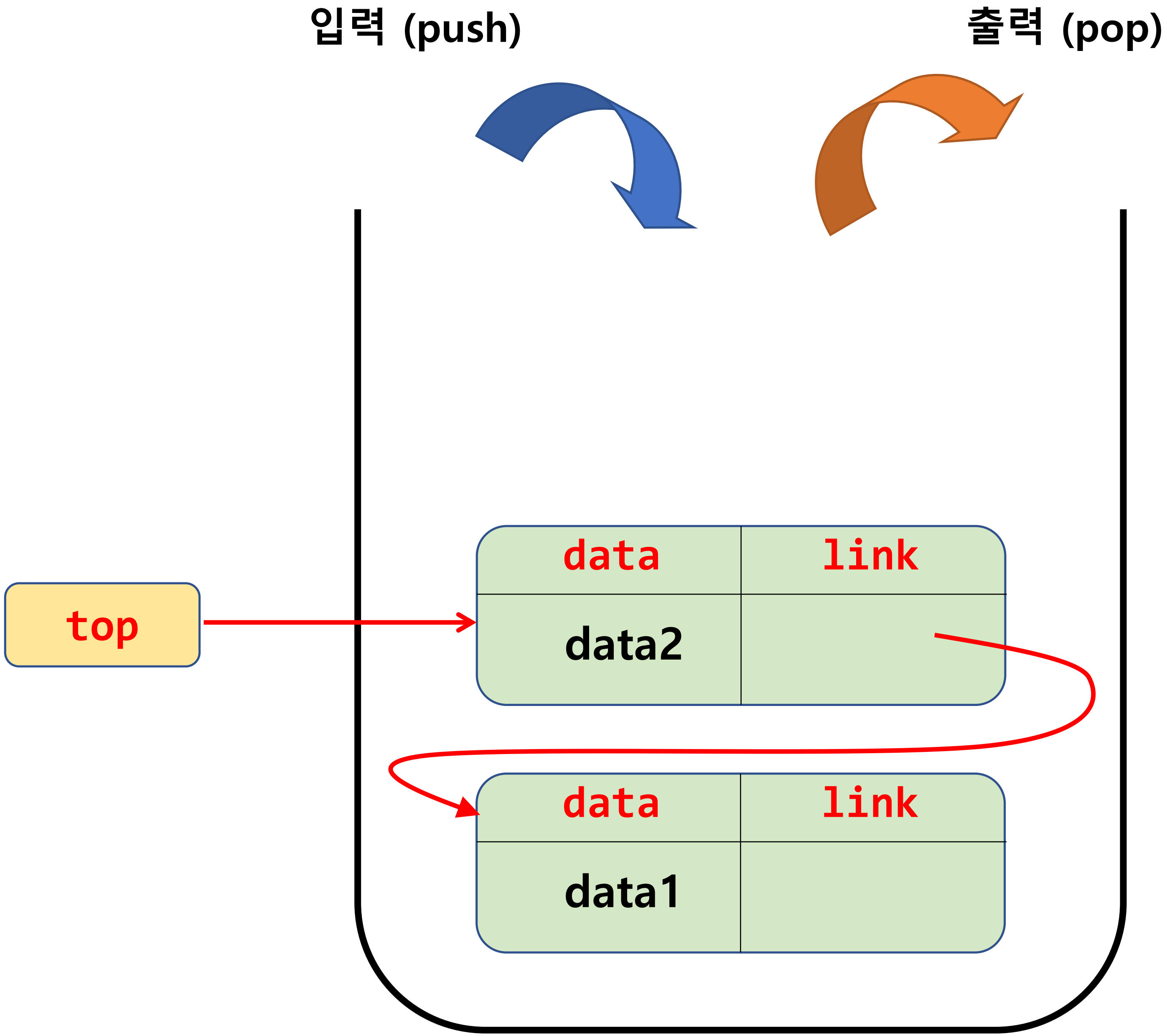
push(data2);

push(data3);

pop();

Data Struct

Stack



push(data1);

push(data2);

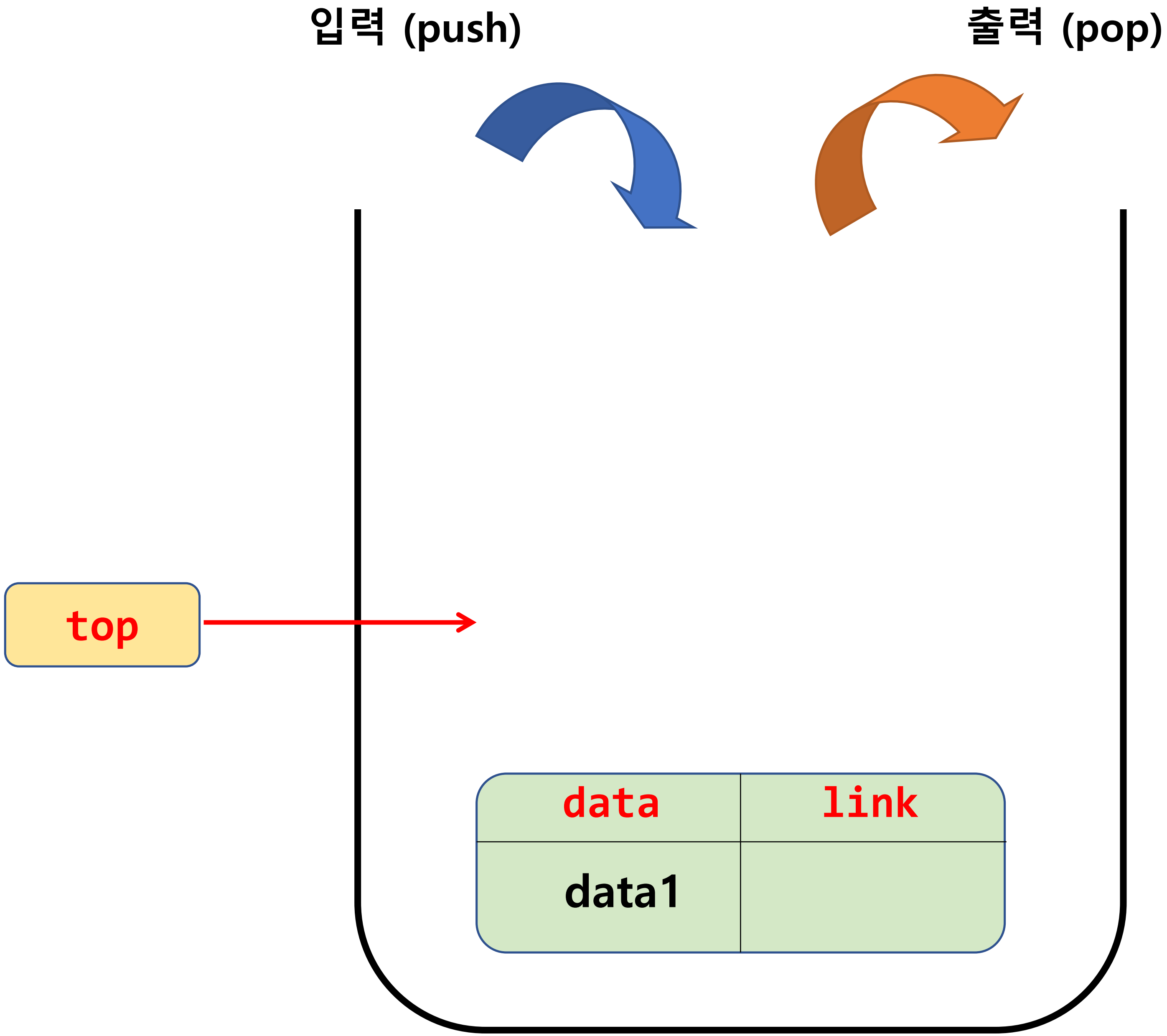
push(data3);

pop();

pop();

Data Struct

Stack



입력 (push)

출력 (pop)

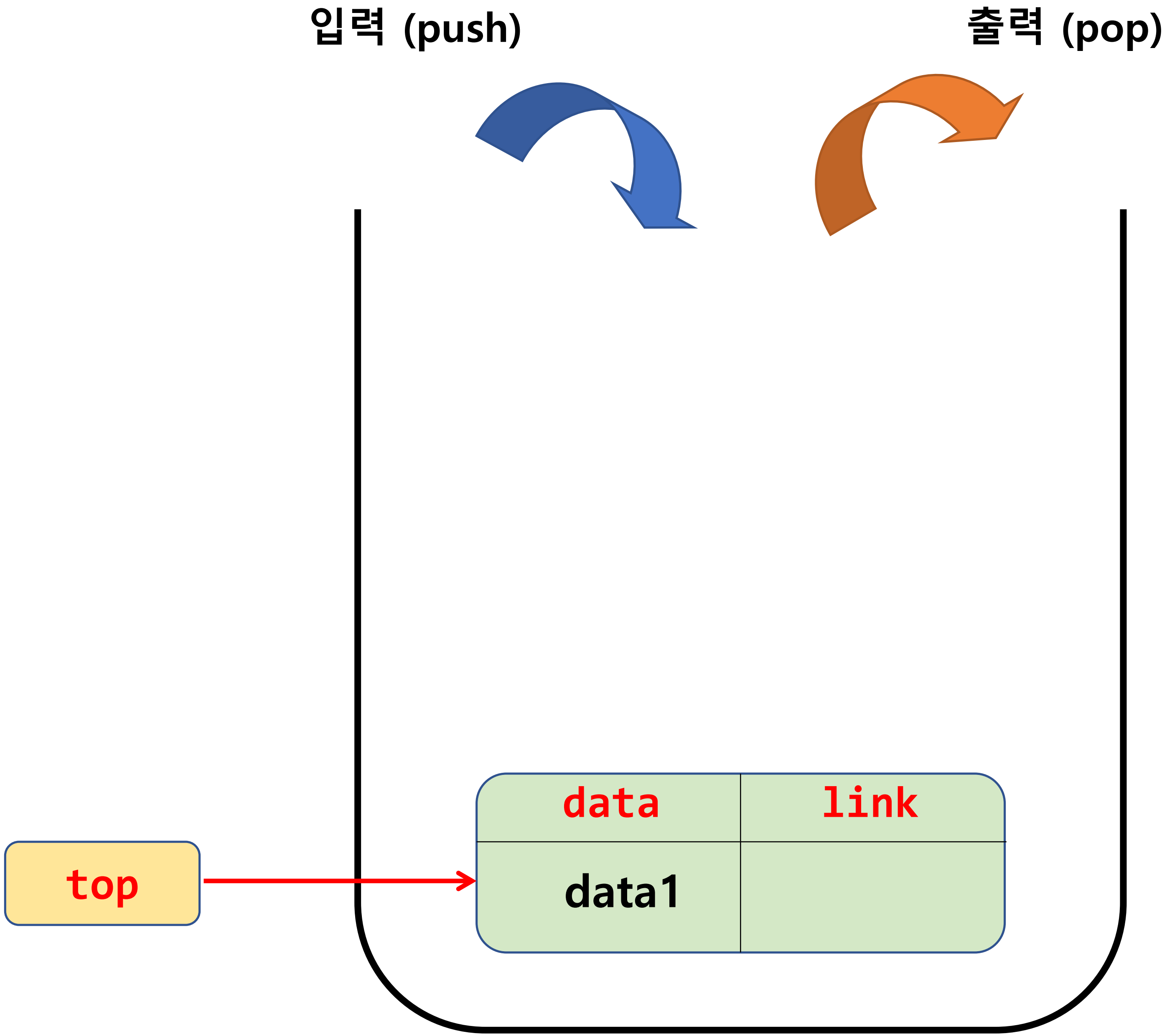
top

data	link
data1	

```
push(data1);

push(data2);

push(data3);


pop();

pop();
```

Data Struct

Stack



입력 (push)

출력 (pop)

data

link

top

data1

push(data1);

push(data2);

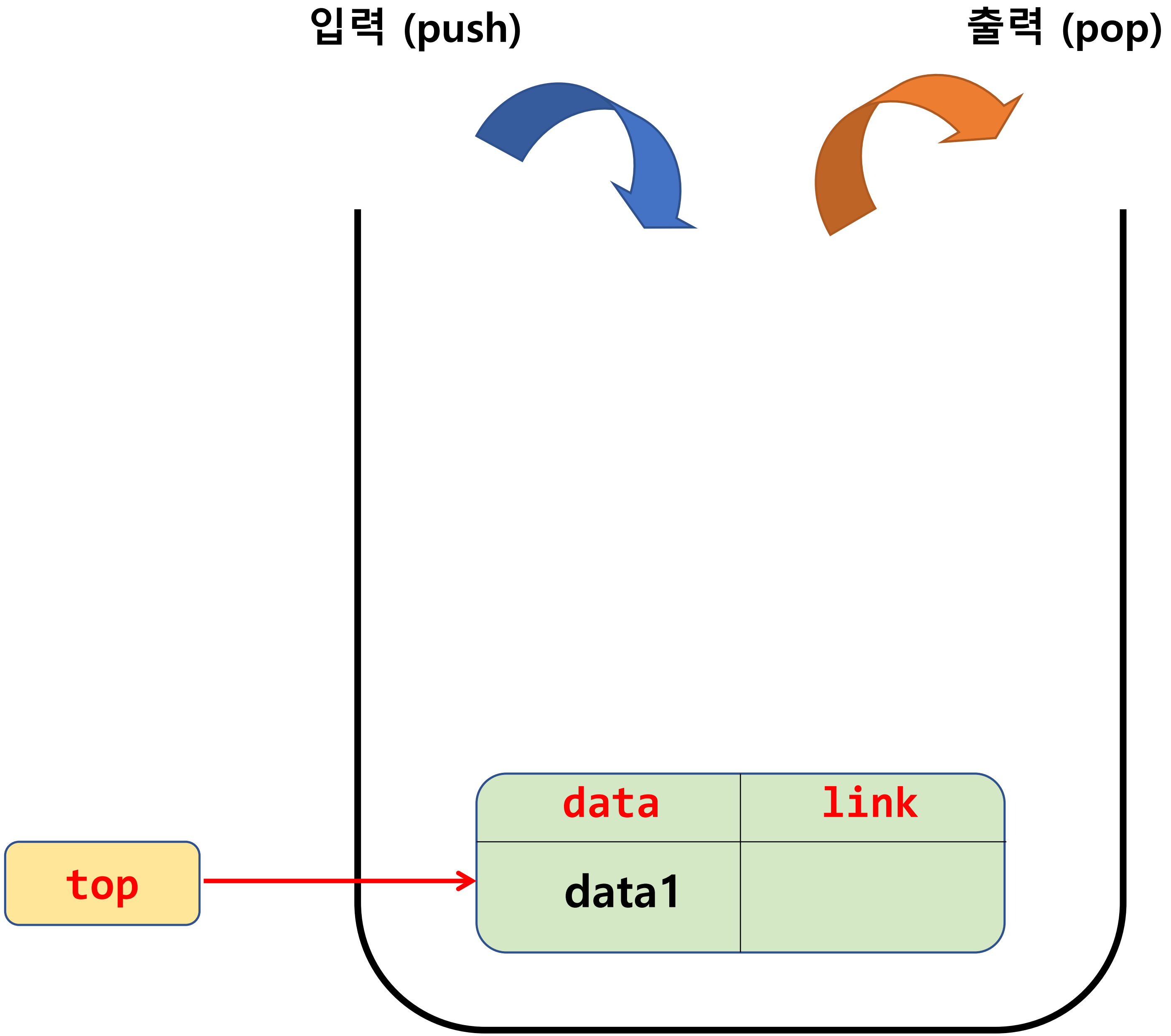
push(data3);

pop();

pop();

Data Struct

Stack



push(data1);

push(data2);

push(data3);

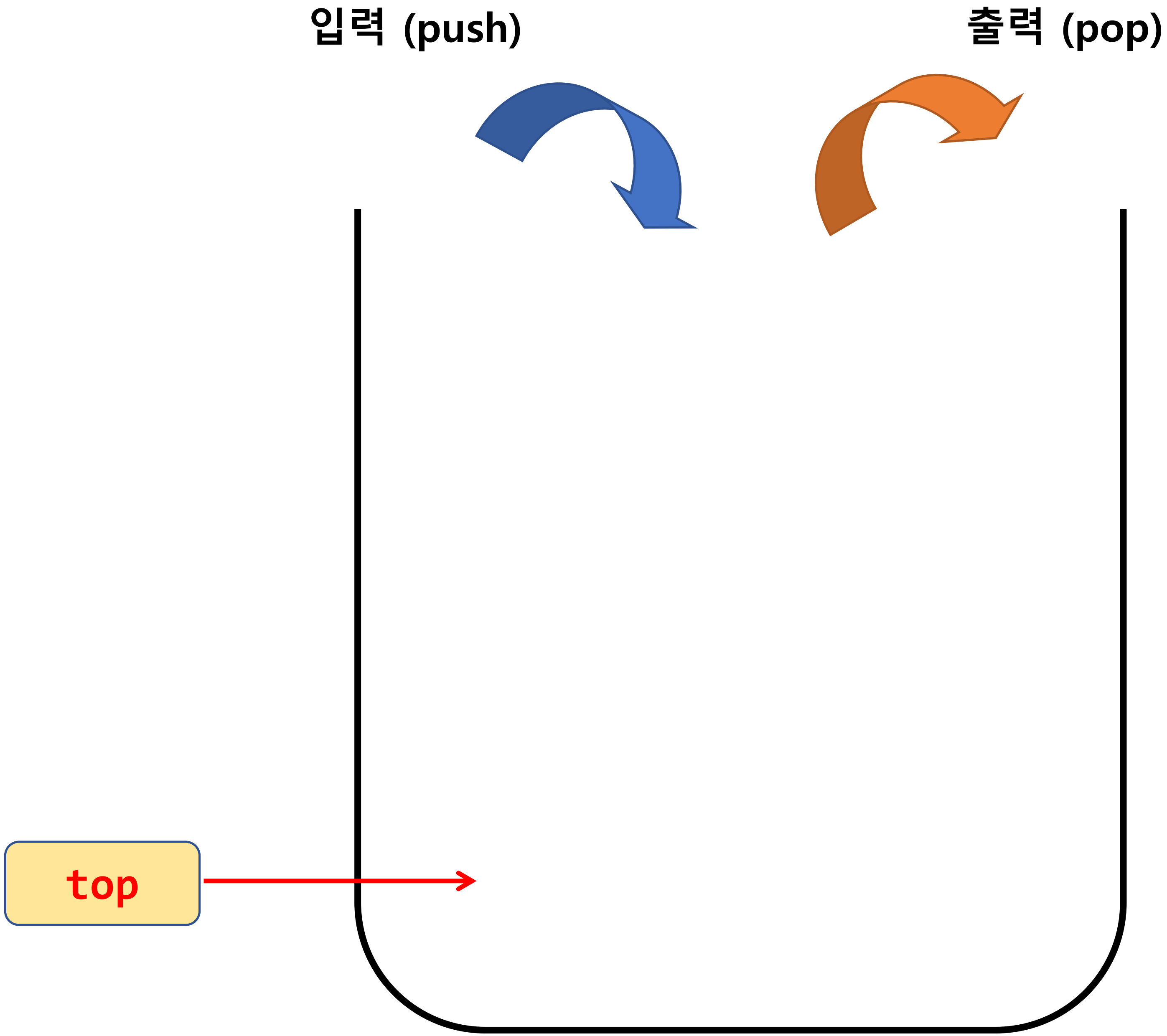
pop();

pop();

pop();

Data Struct

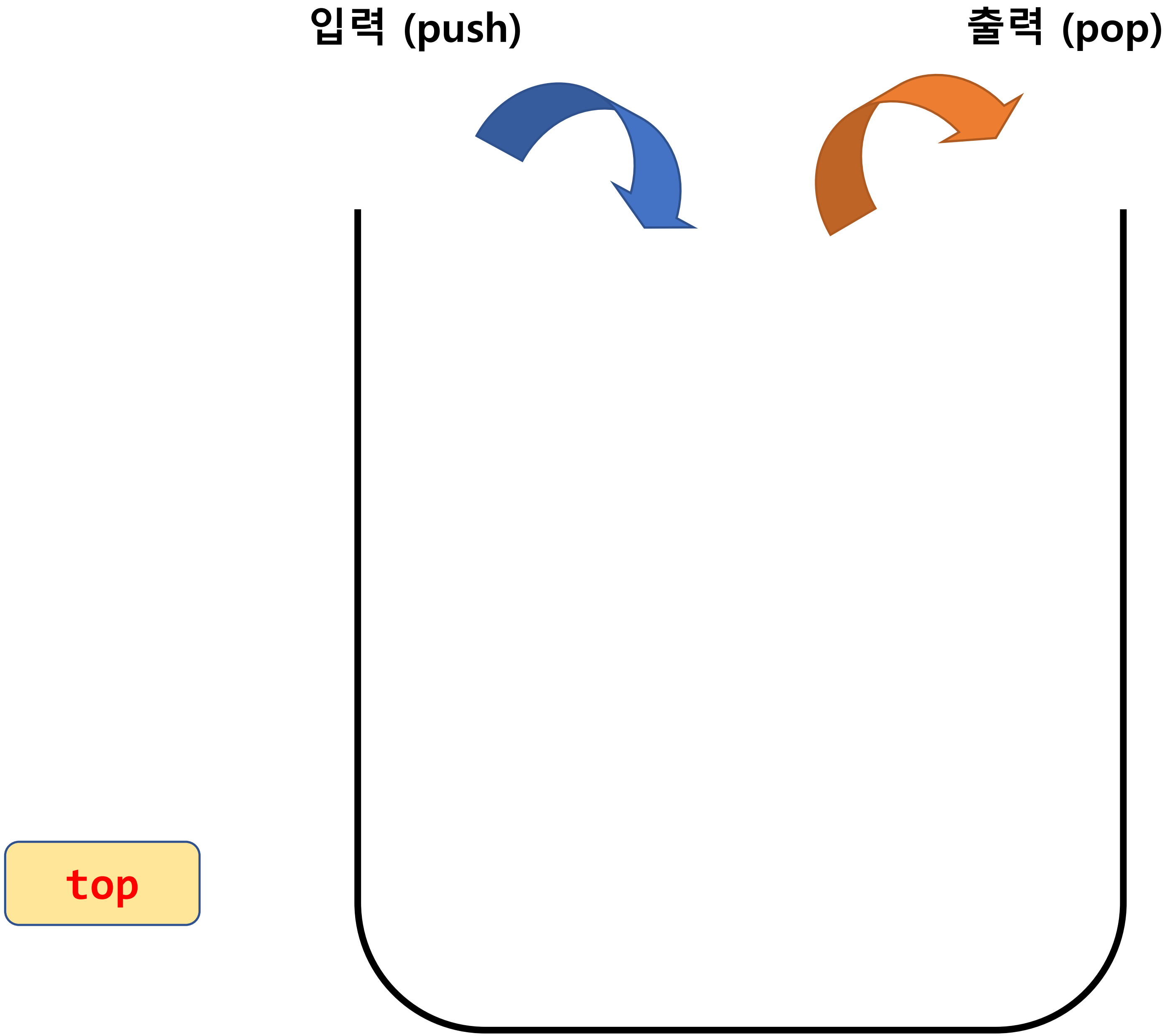
Stack



```
push(data1);

push(data2);

push(data3);


pop();

pop();

pop();
```

Data Struct

Stack



push(data1);

push(data2);

push(data3);

pop();

pop();

pop();

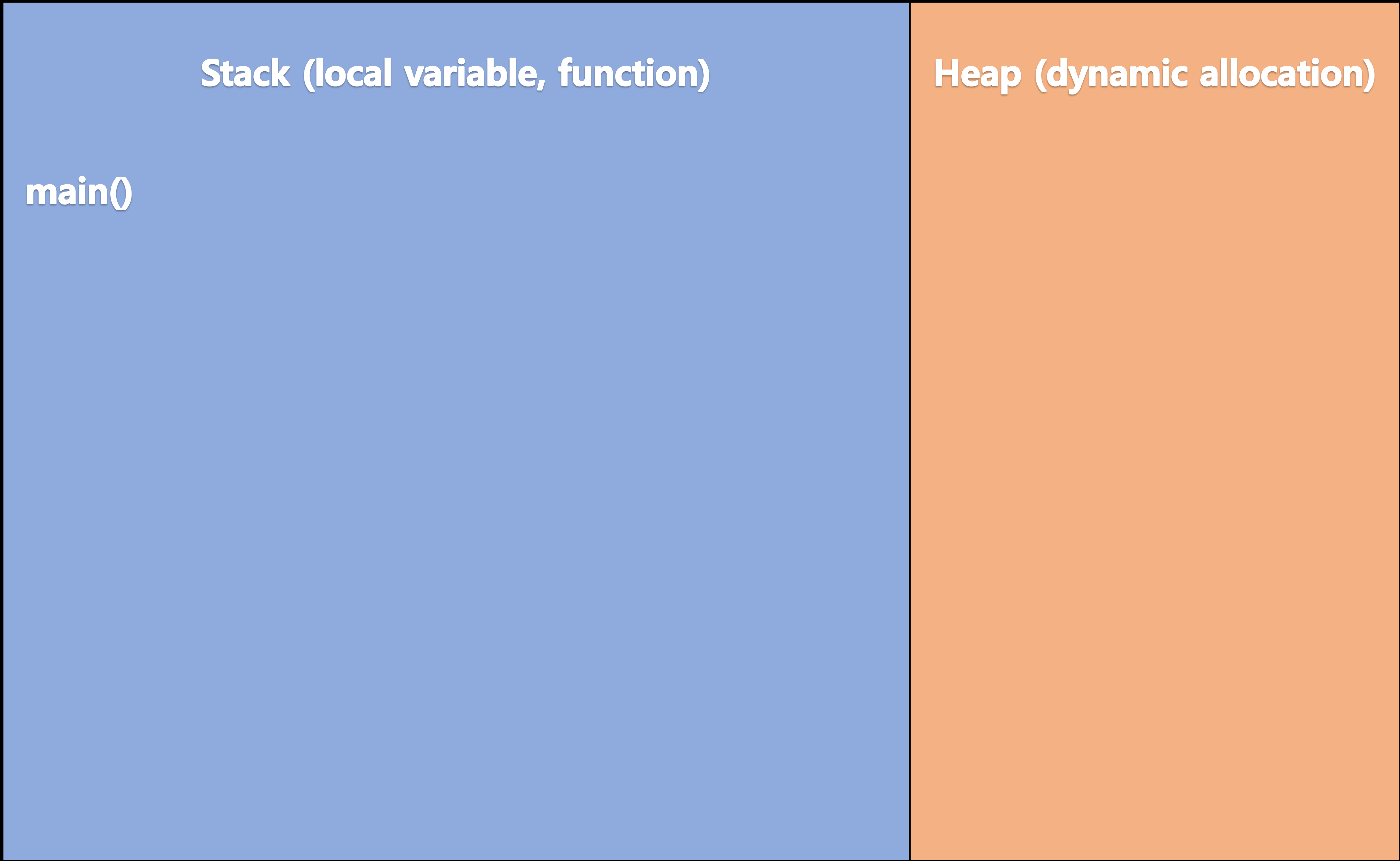
Stack 코드 분석 : push()

Data Struct

Stack

code)

```
Stack* top = NULL;
```



Stack (local variable, function)

main()

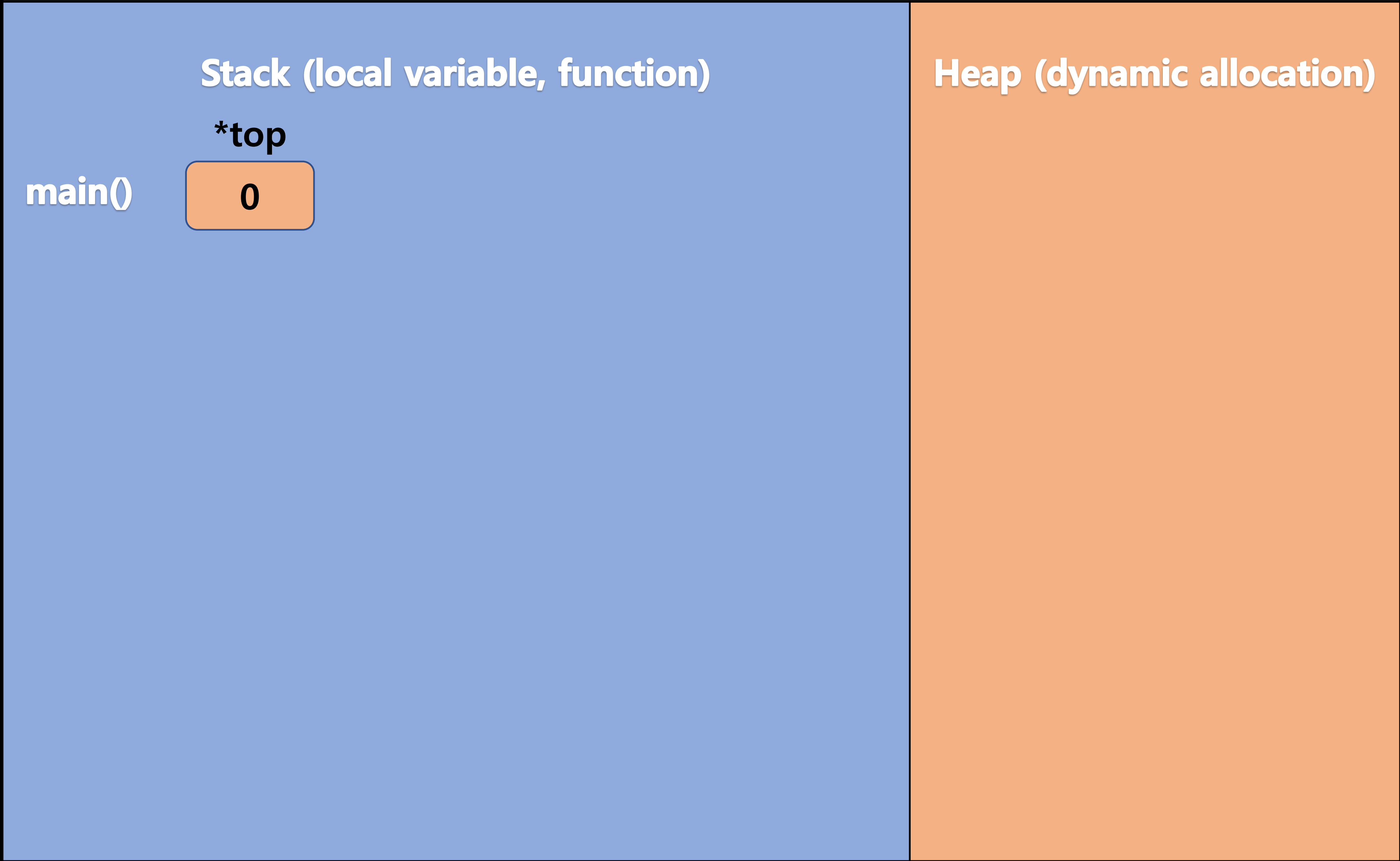
Heap (dynamic allocation)

Data Struct

Stack

code)

```
Stack* top = NULL;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

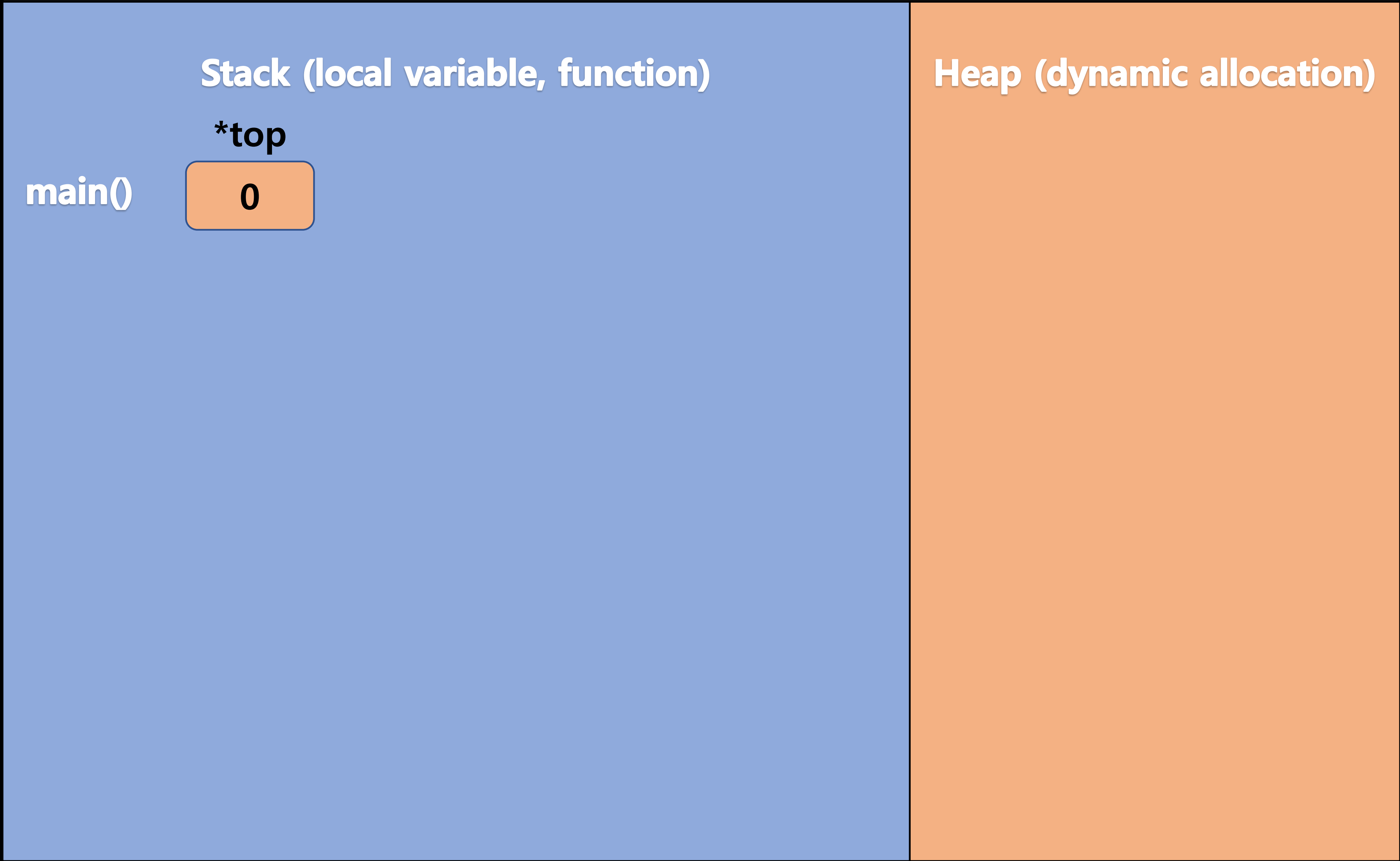
0

Data Struct

Stack

code)

```
push(&top, 10);
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

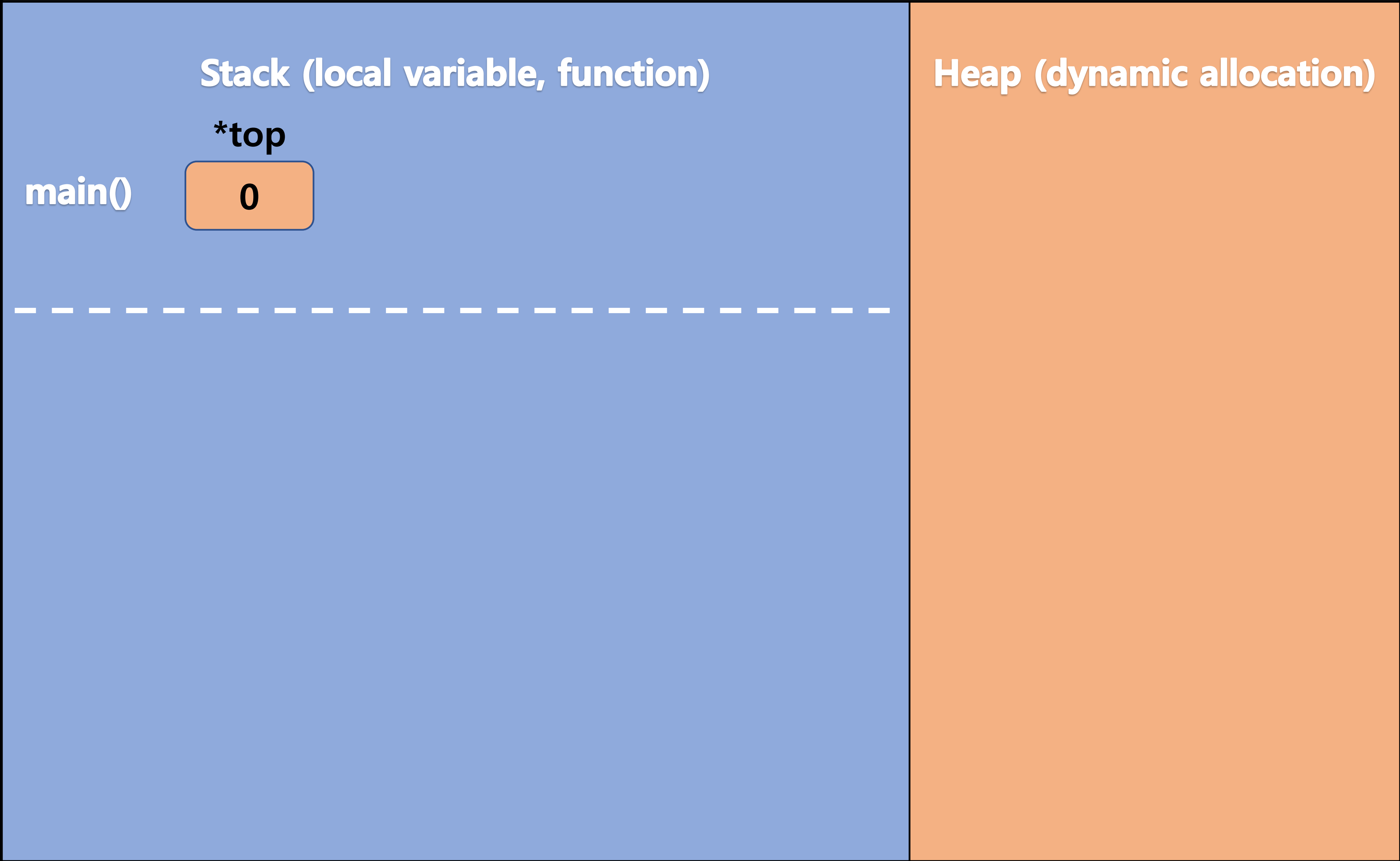
0

Data Struct

Stack

code)

```
push(&top, 10);
```



Stack (local variable, function)

main()

*top

0

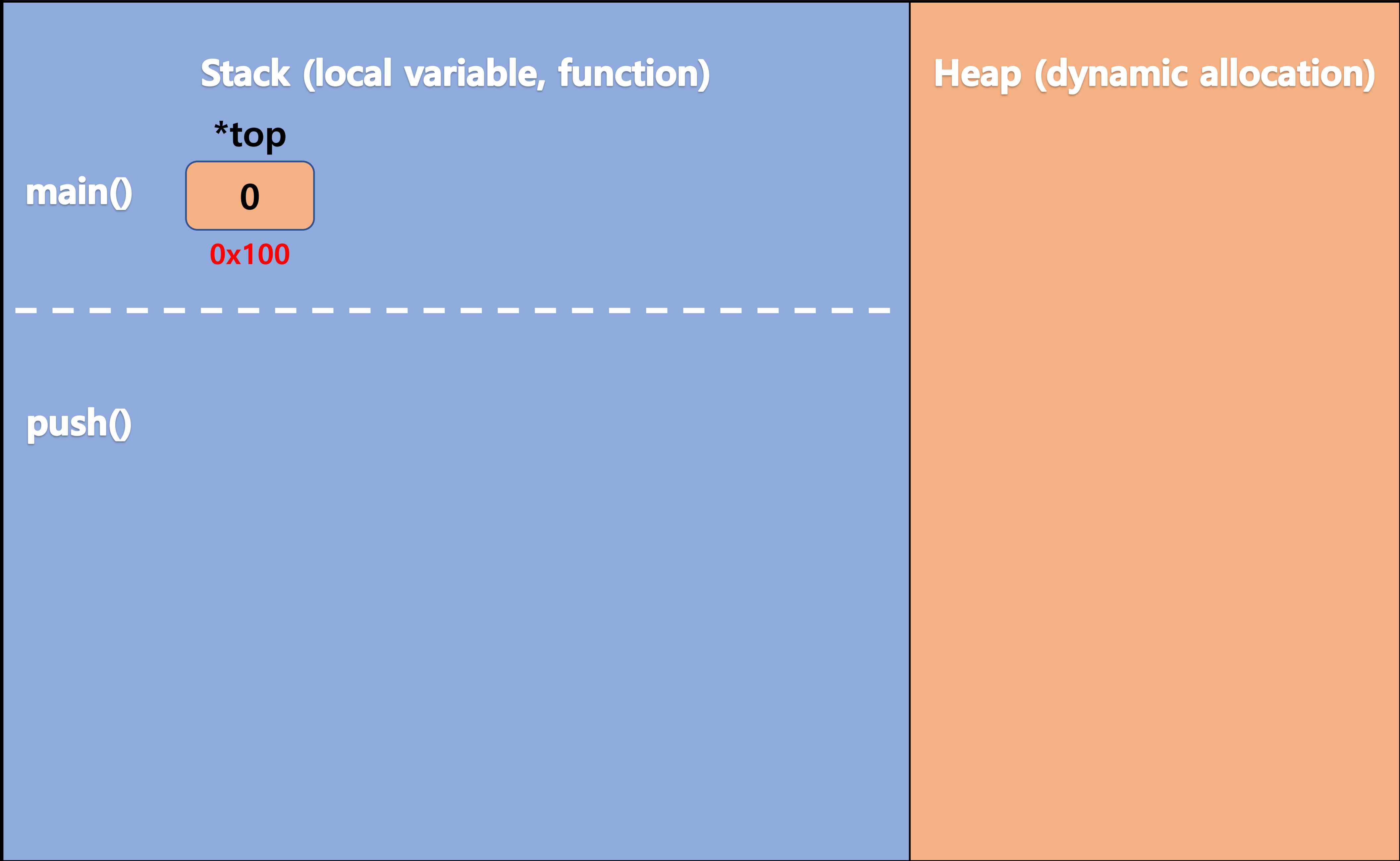
Heap (dynamic allocation)

Data Struct

Stack

code)

```
push(&top, 10);
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0

0x100

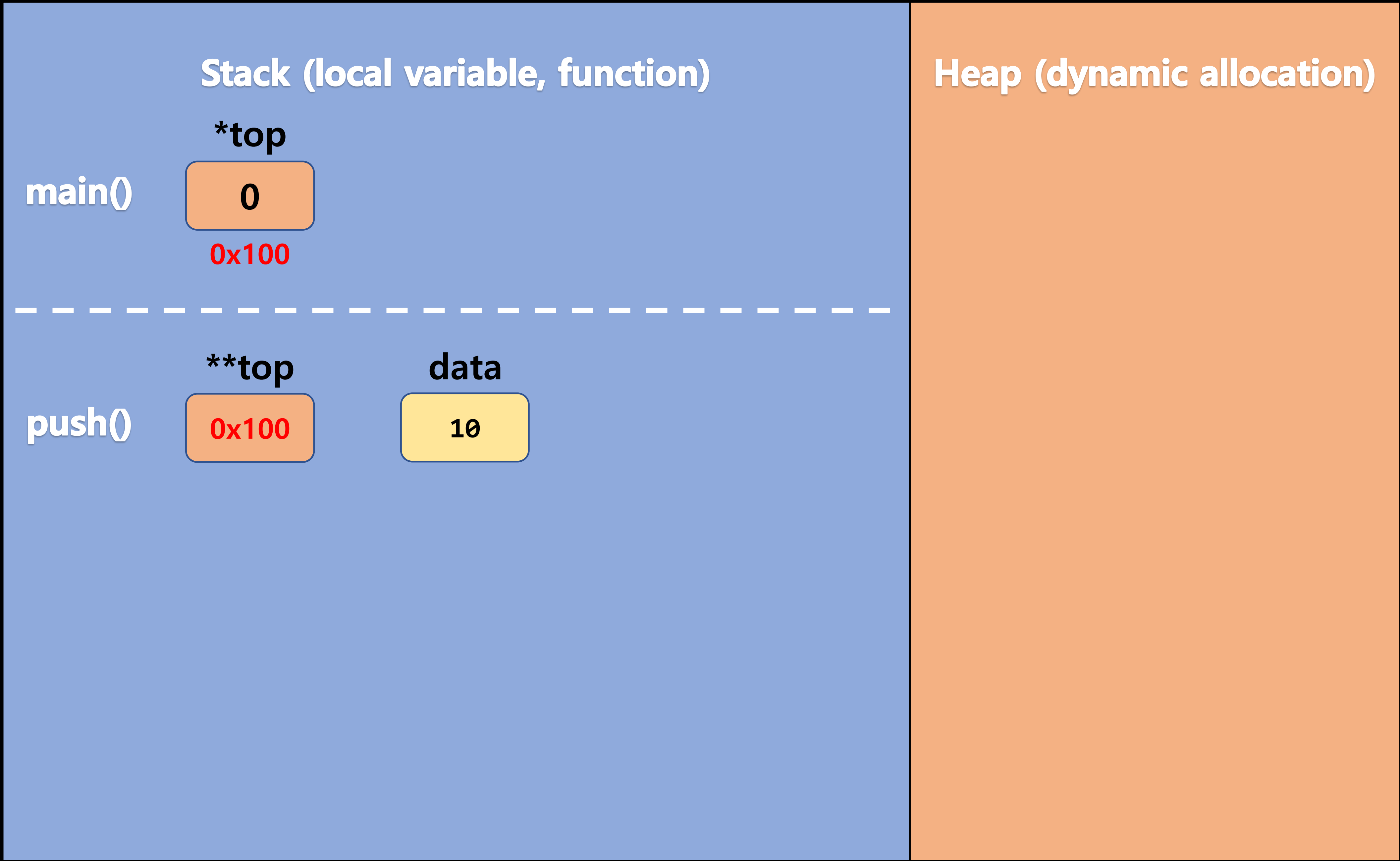
push()

Data Struct

Stack

code)

```
push(&top, 10);
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0

0x100

push()

**top

0x100

data

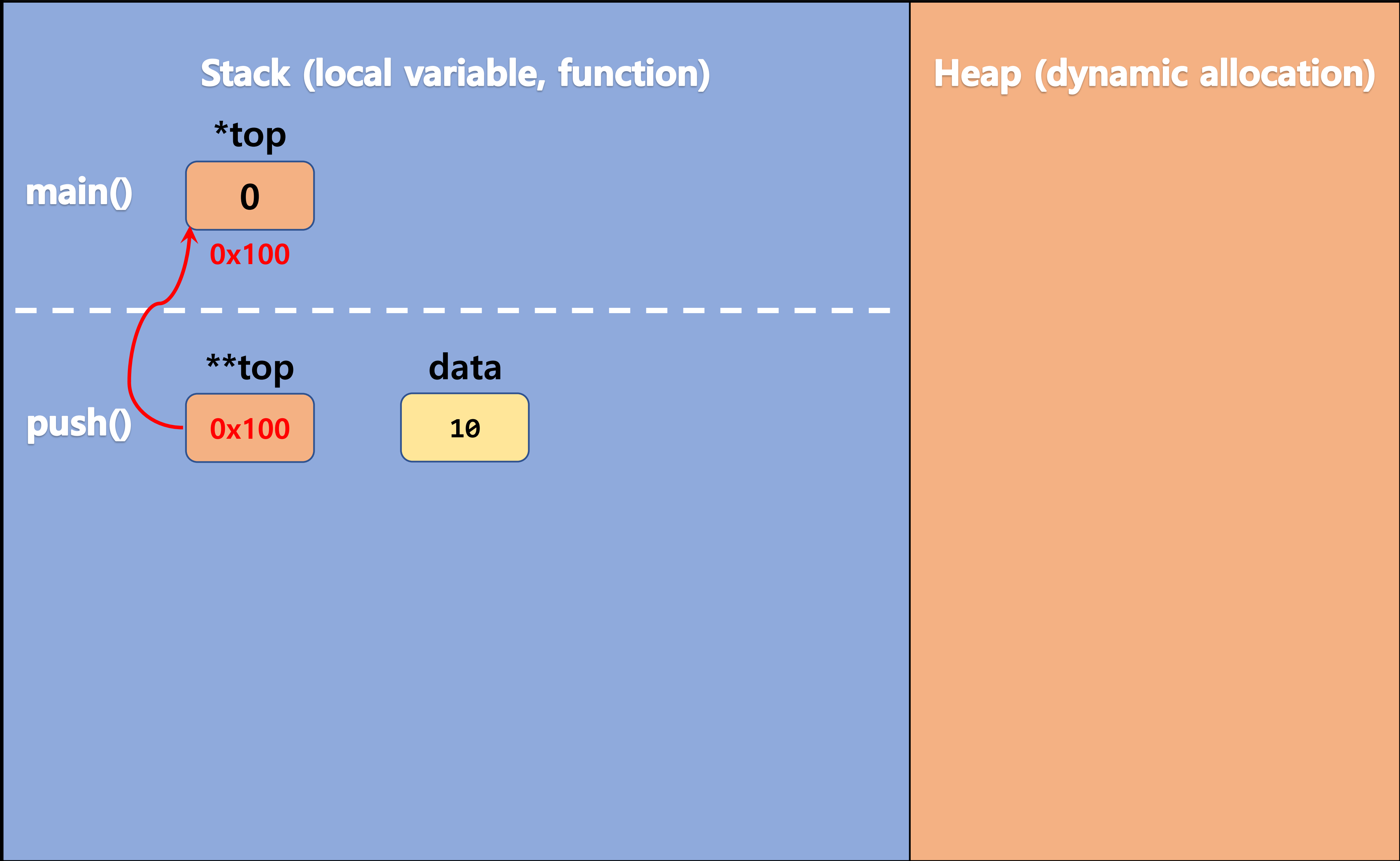
10

Data Struct

Stack

code)

```
push(&top, 10);
```

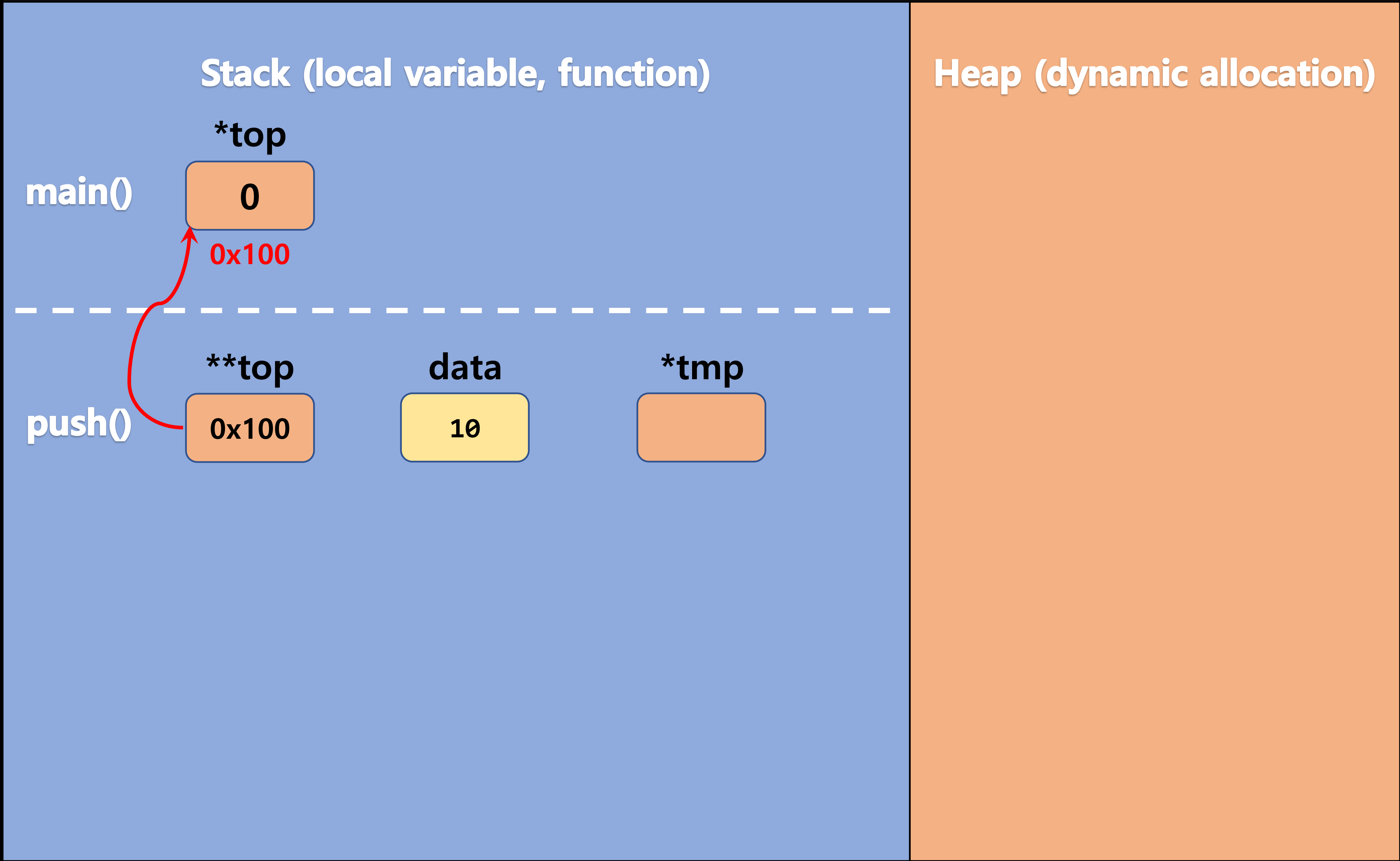


Data Struct

Stack

code)

```
Stack* tmp = *top;
```

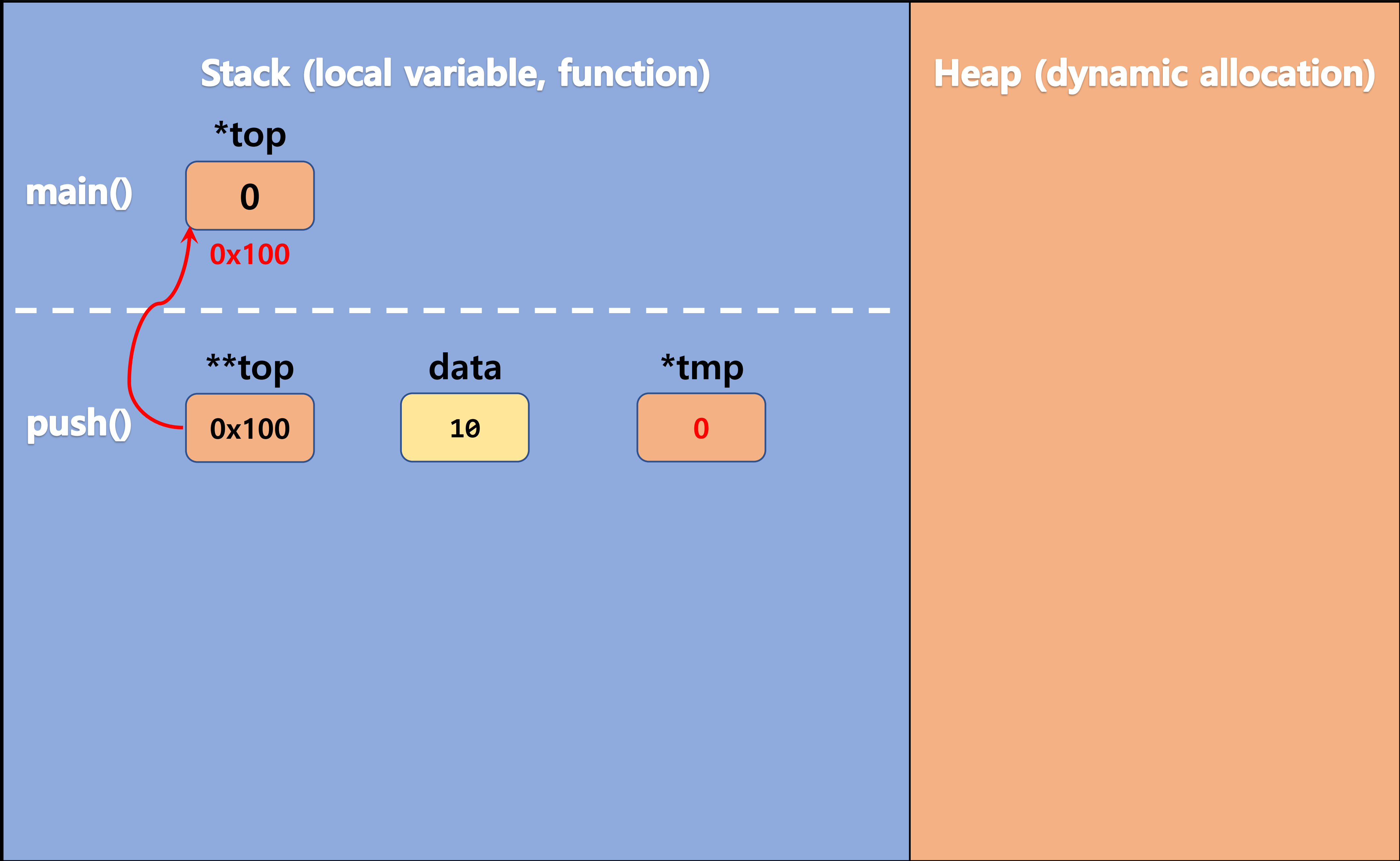


Data Struct

Stack

code)

```
Stack* tmp = *top;
```

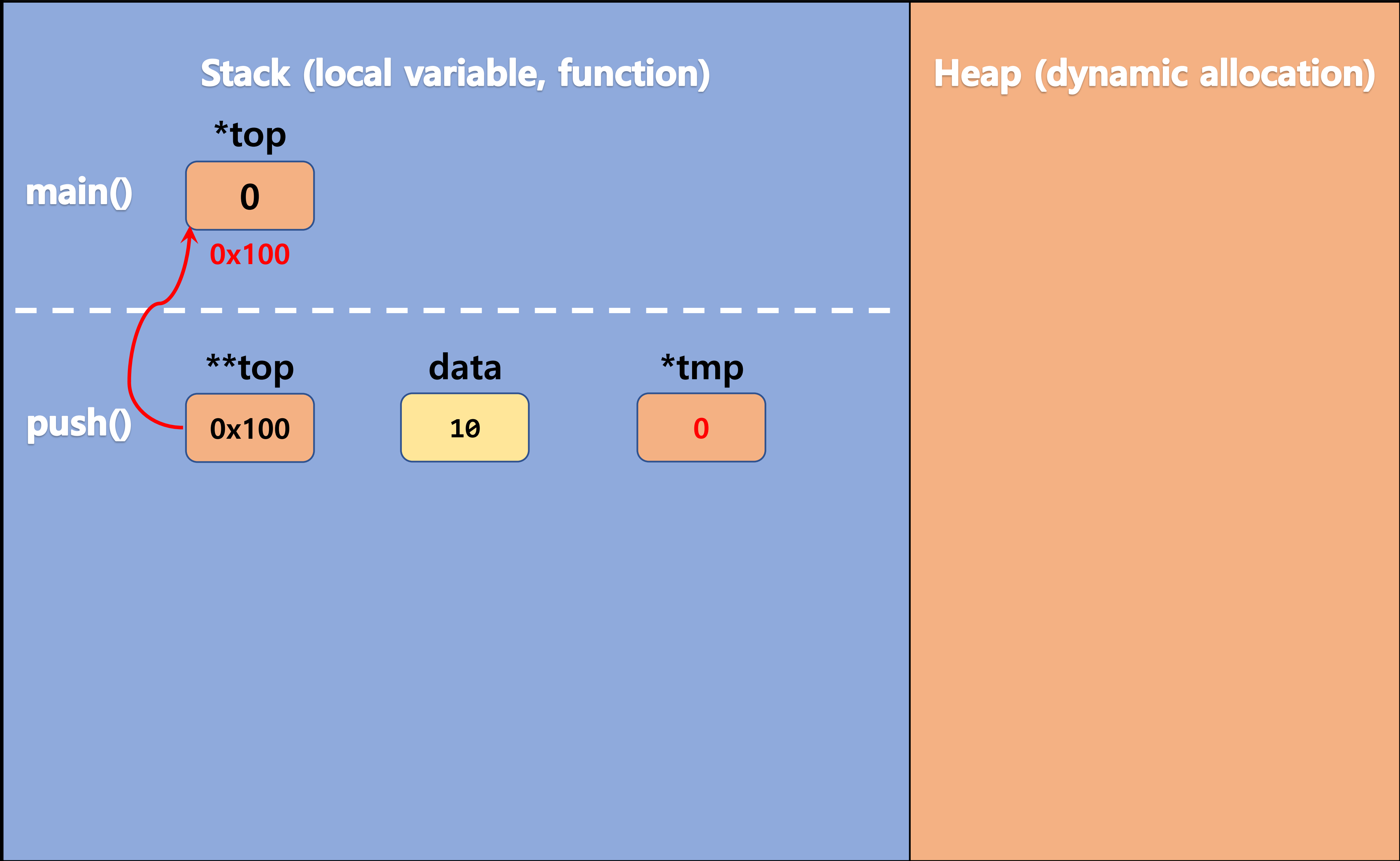


Data Struct

Stack

code)

```
*top = getNode();
```

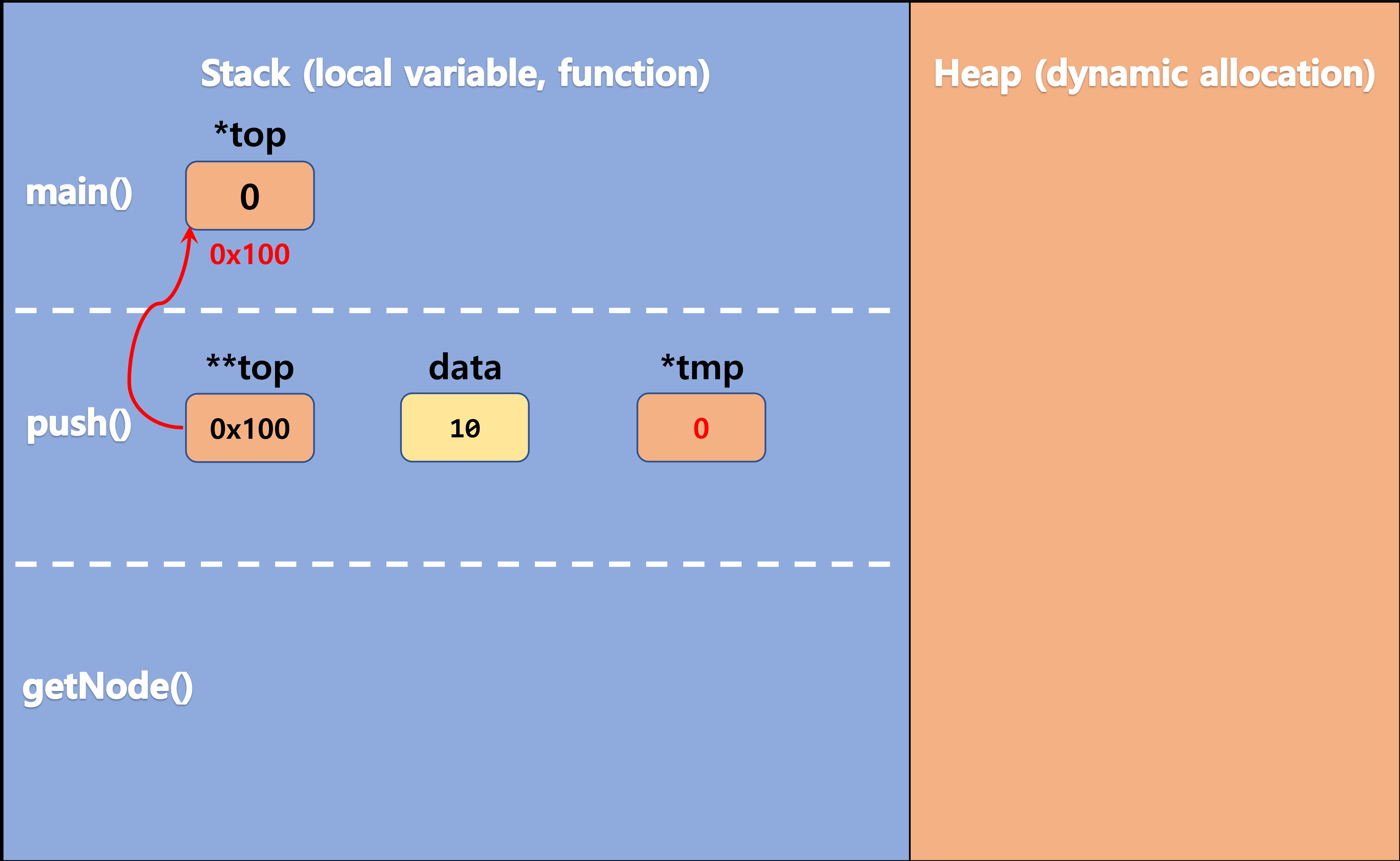


Data Struct

Stack

code)

```
*top = getNode();
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0

0x100

push()

**top

0x100

data

10

*tmp

0

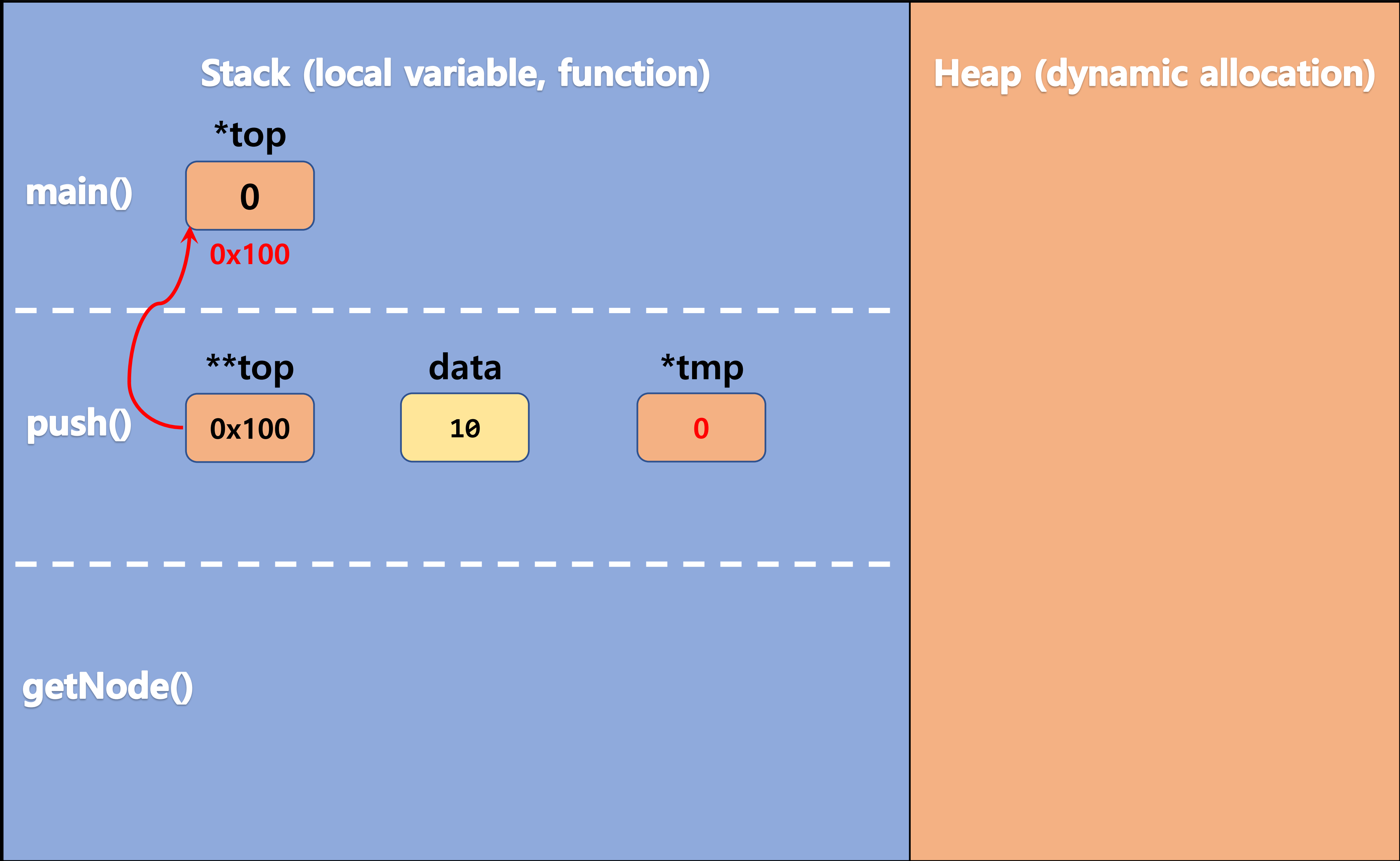
getNode()

Data Struct

Stack

code)

```
Stack* newNode = (Stack*)malloc(sizeof(Stack));
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0

0x100

push()

**top

0x100

data

10

*tmp

0

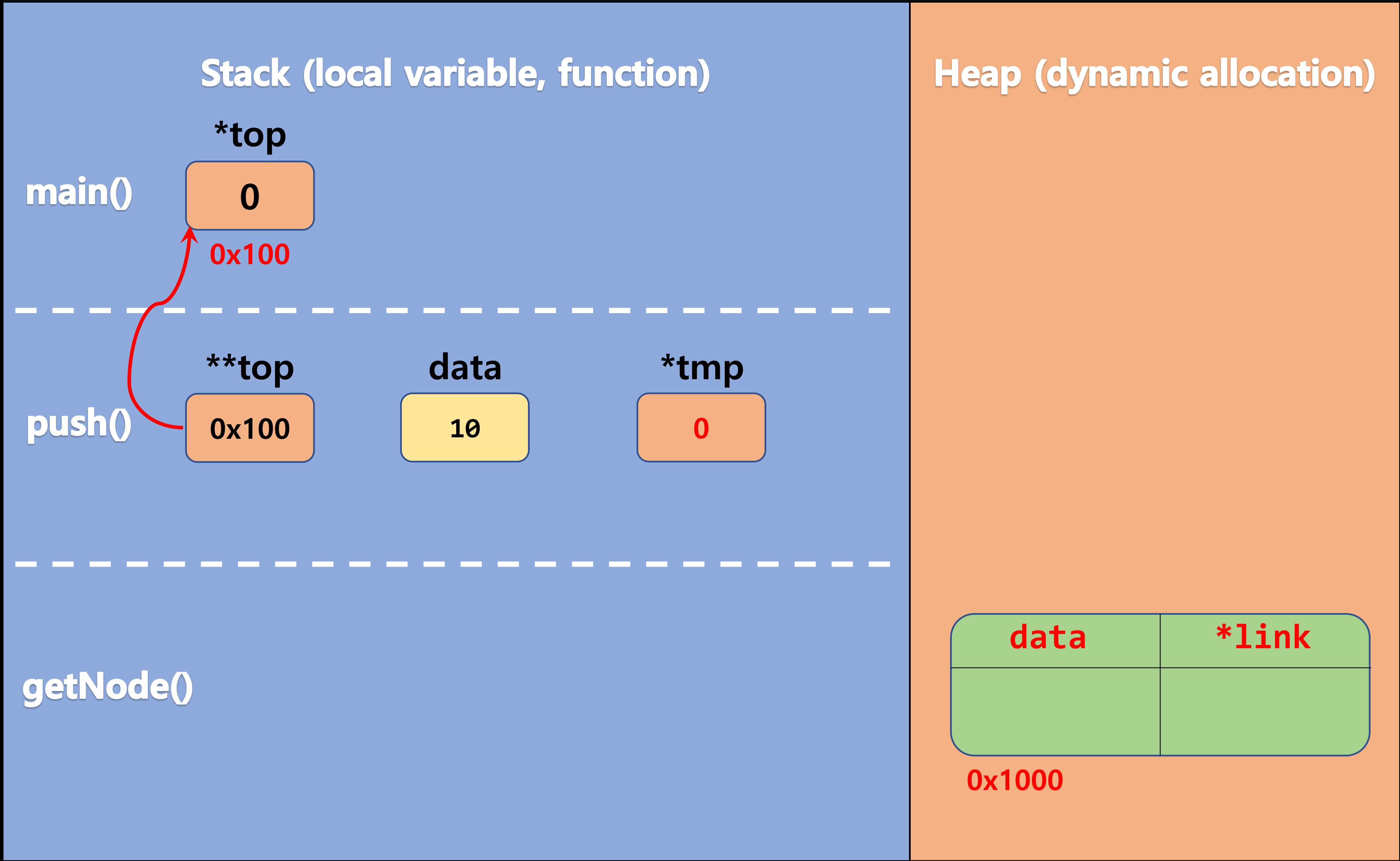
getNode()

Data Struct

Stack

code)

```
Stack* newNode = (Stack*)malloc(sizeof(Stack));
```

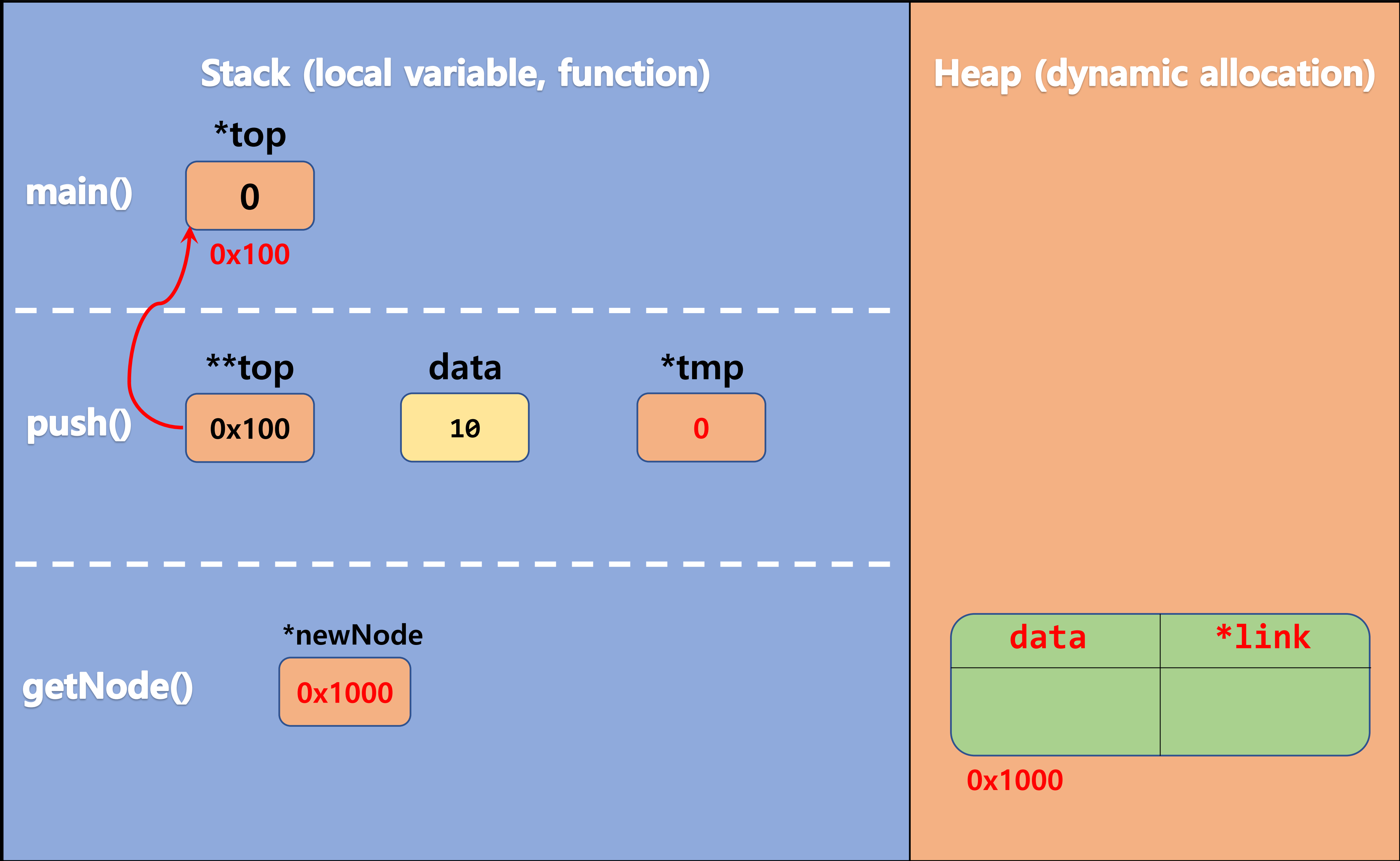


Data Struct

Stack

code)

```
Stack* newNode = (Stack*)malloc(sizeof(Stack));
```

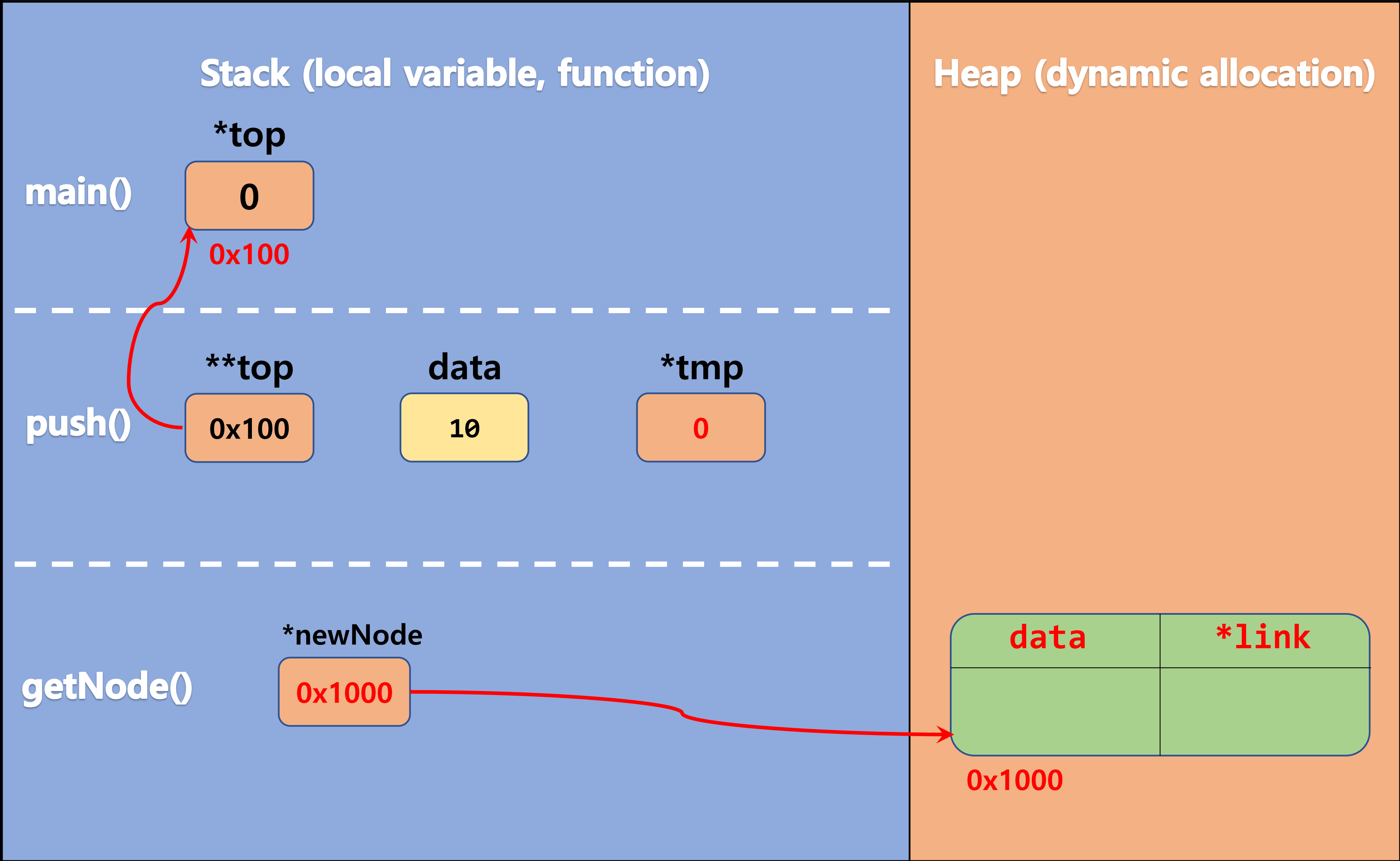


Data Struct

Stack

code)

```
Stack* newNode = (Stack*)malloc(sizeof(Stack));
```

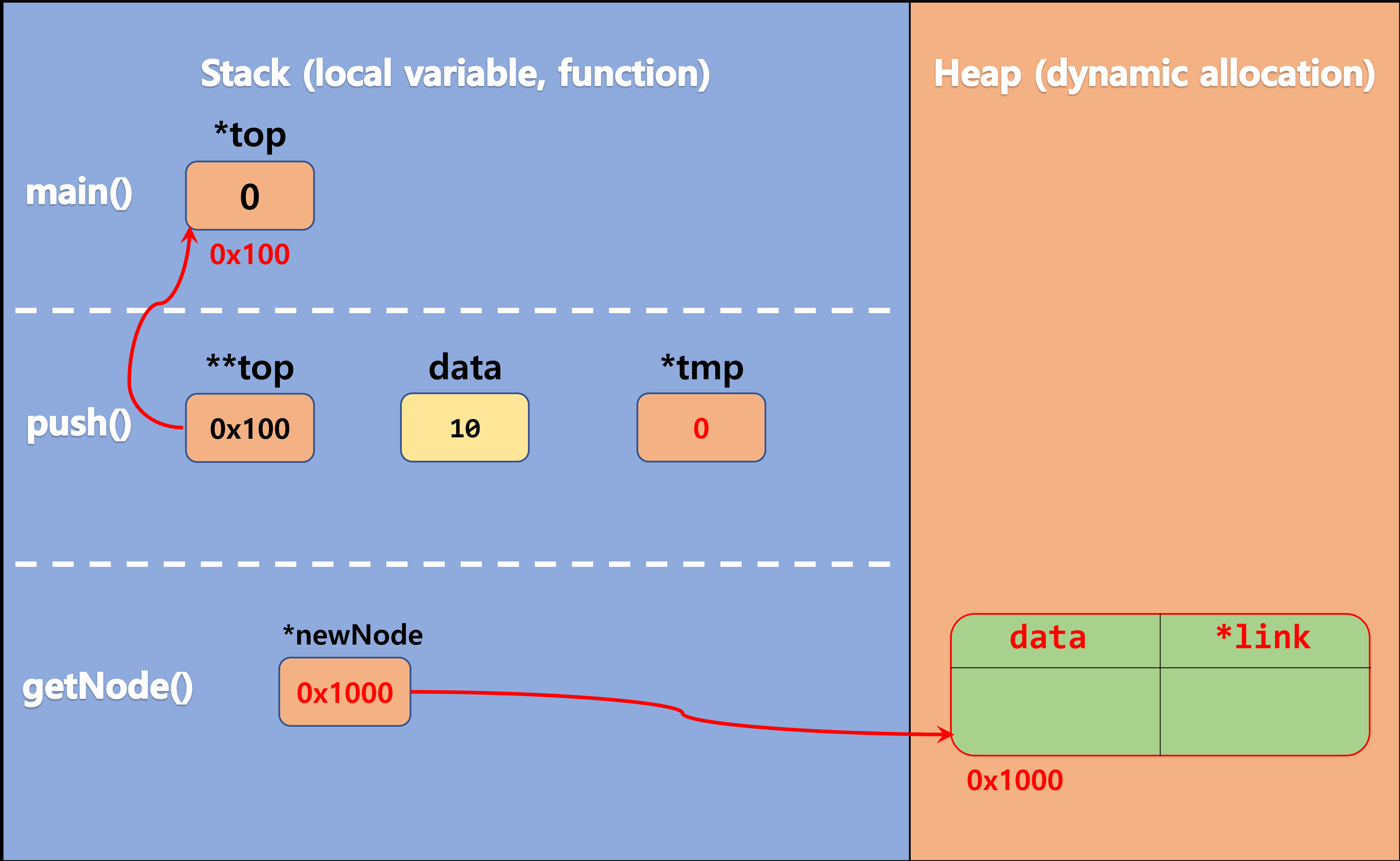


Data Struct

Stack

code)

```
newNode->link = NULL;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0

0x100

push()

**top

0x100

data

10

*tmp

0

getNode()

*newNode

0x1000

data

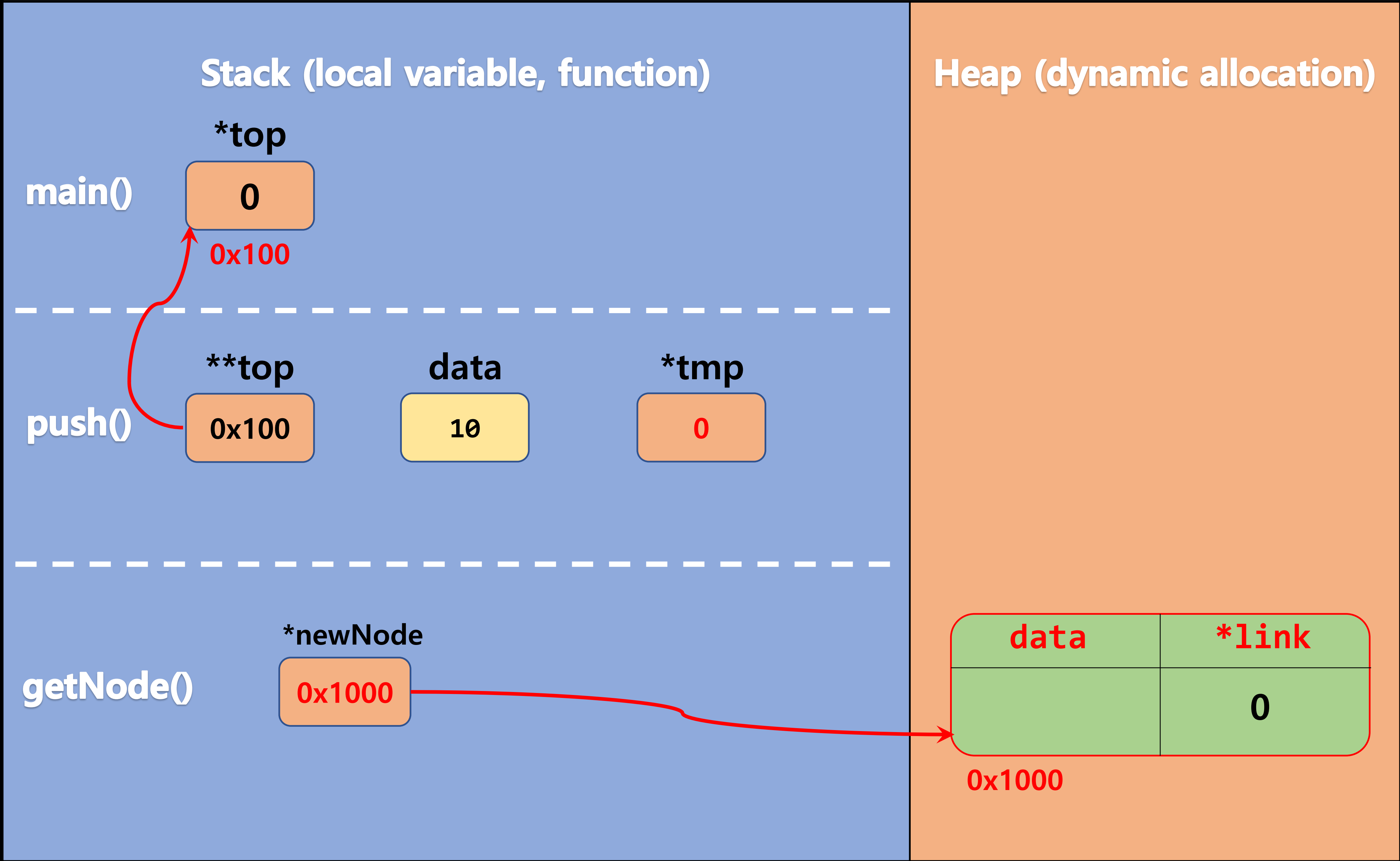
*link

0x1000

Data Struct Stack

code)

```
newNode->link = NULL;
```

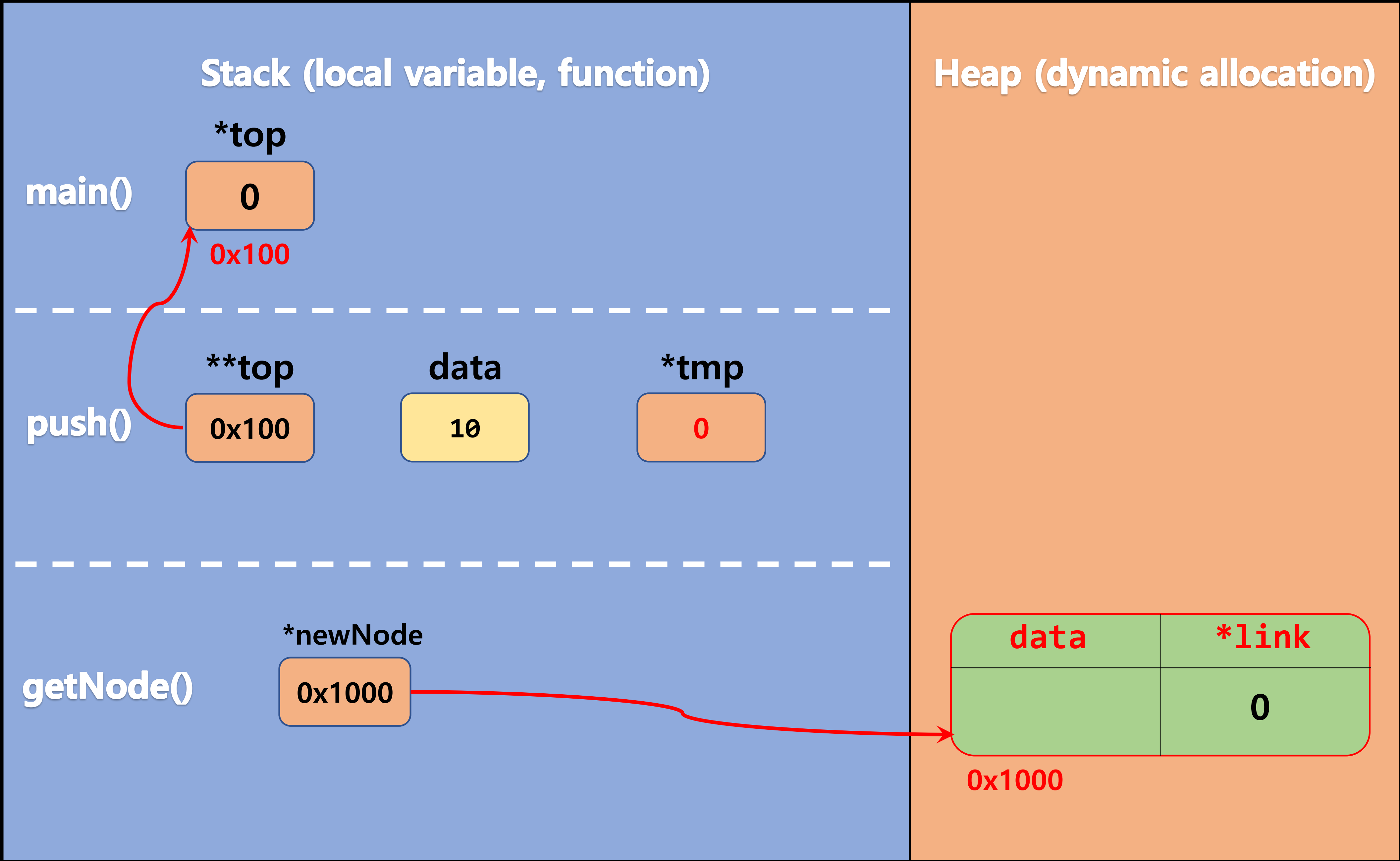


Data Struct

Stack

code)

```
return newNode;
```

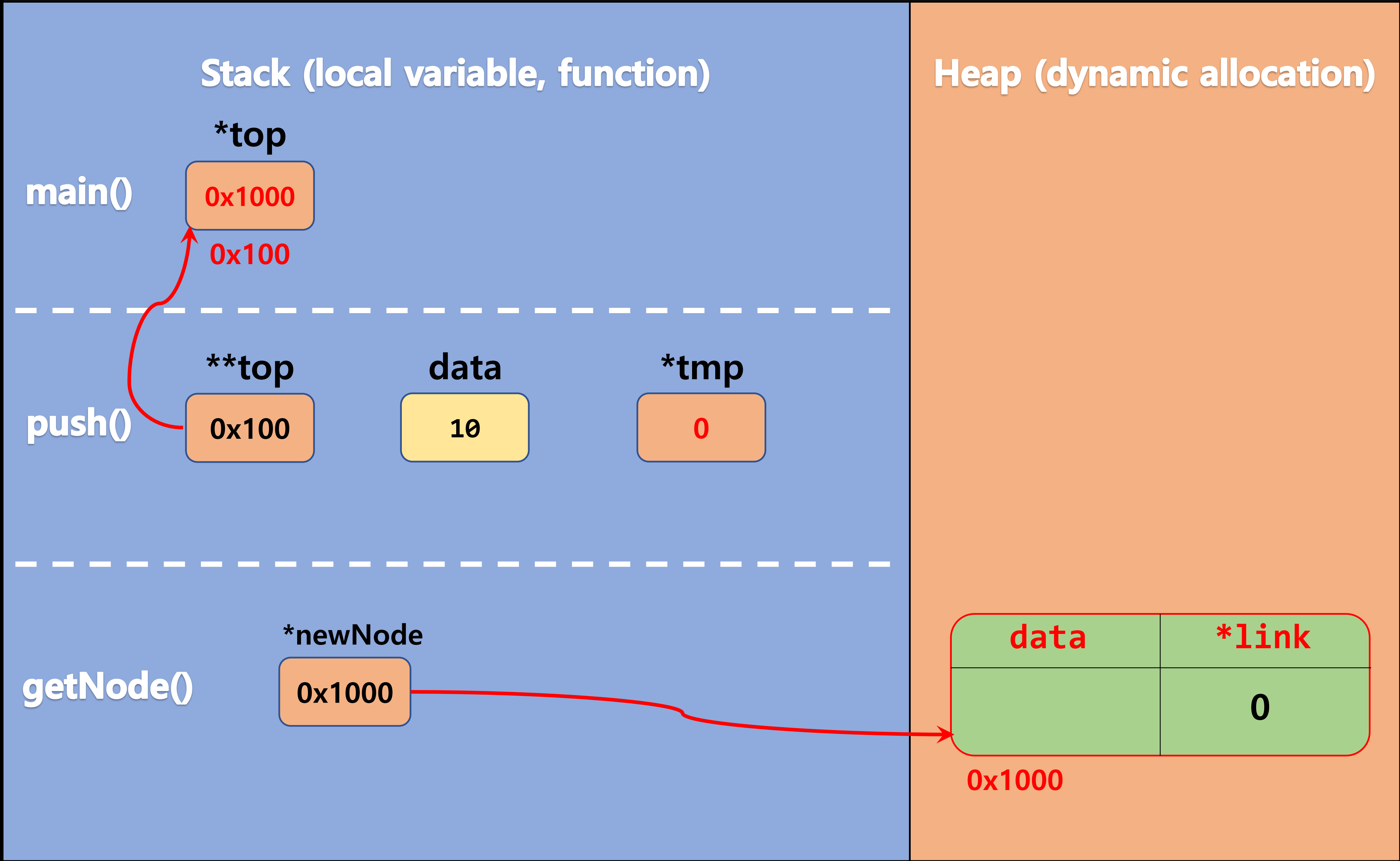


Data Struct

Stack

code)

```
return newNode;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x1000

0x100

push()

**top

0x100

data

10

*tmp

0

getNode()

*newNode

0x1000

data

*link

0

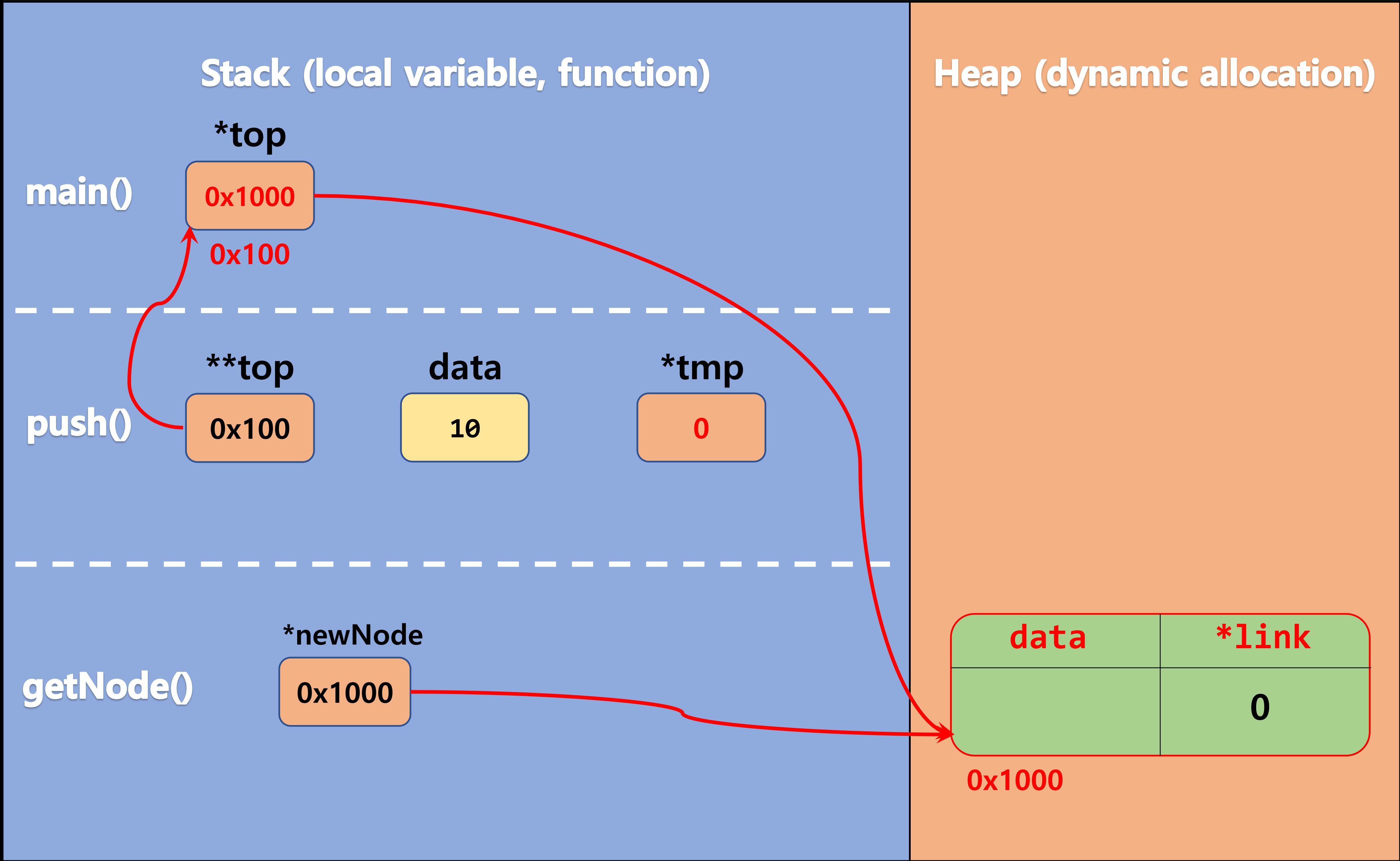
0x1000

Data Struct

Stack

code)

```
return newNode;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x1000

0x100

push()

**top

0x100

data

10

*tmp

0

getNode()

*newNode

0x1000

data

*link

0

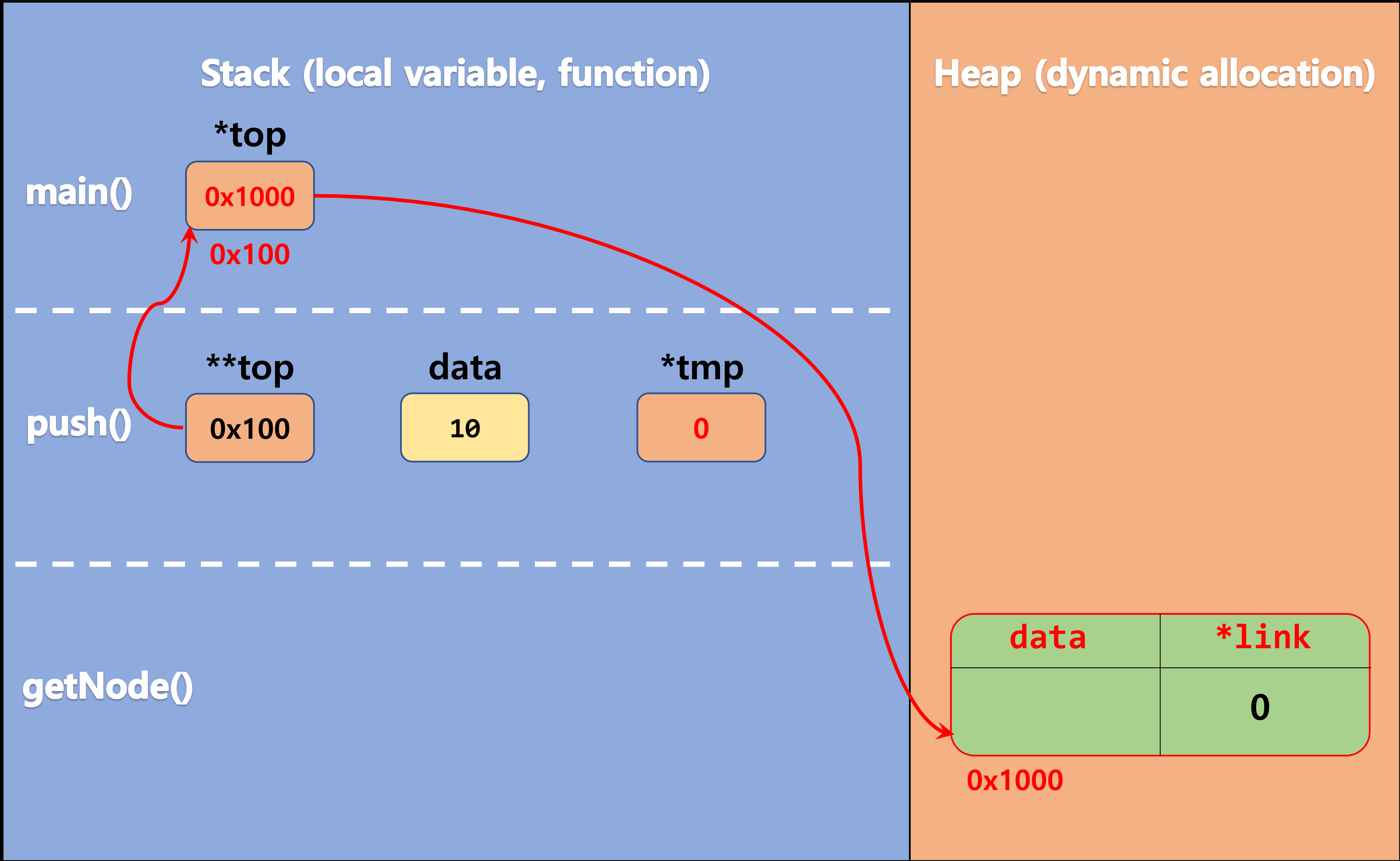
0x1000

Data Struct

Stack

code)

```
return newNode;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x1000

0x100

push()

**top

0x100

data

10

*tmp

0

getNode()

data

*link

0

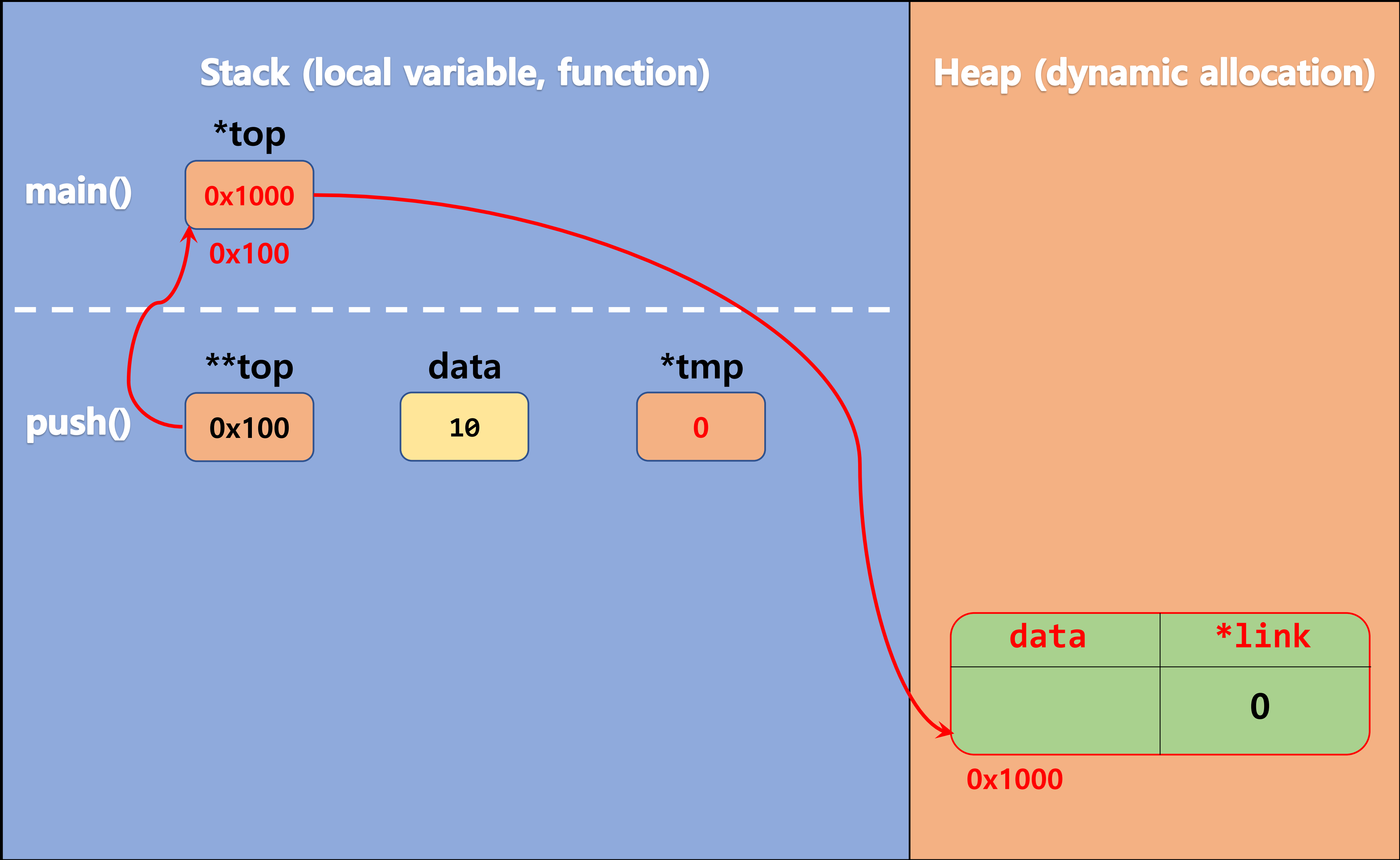
0x1000

code)

```
return newNode;
```

Data Struct

Stack

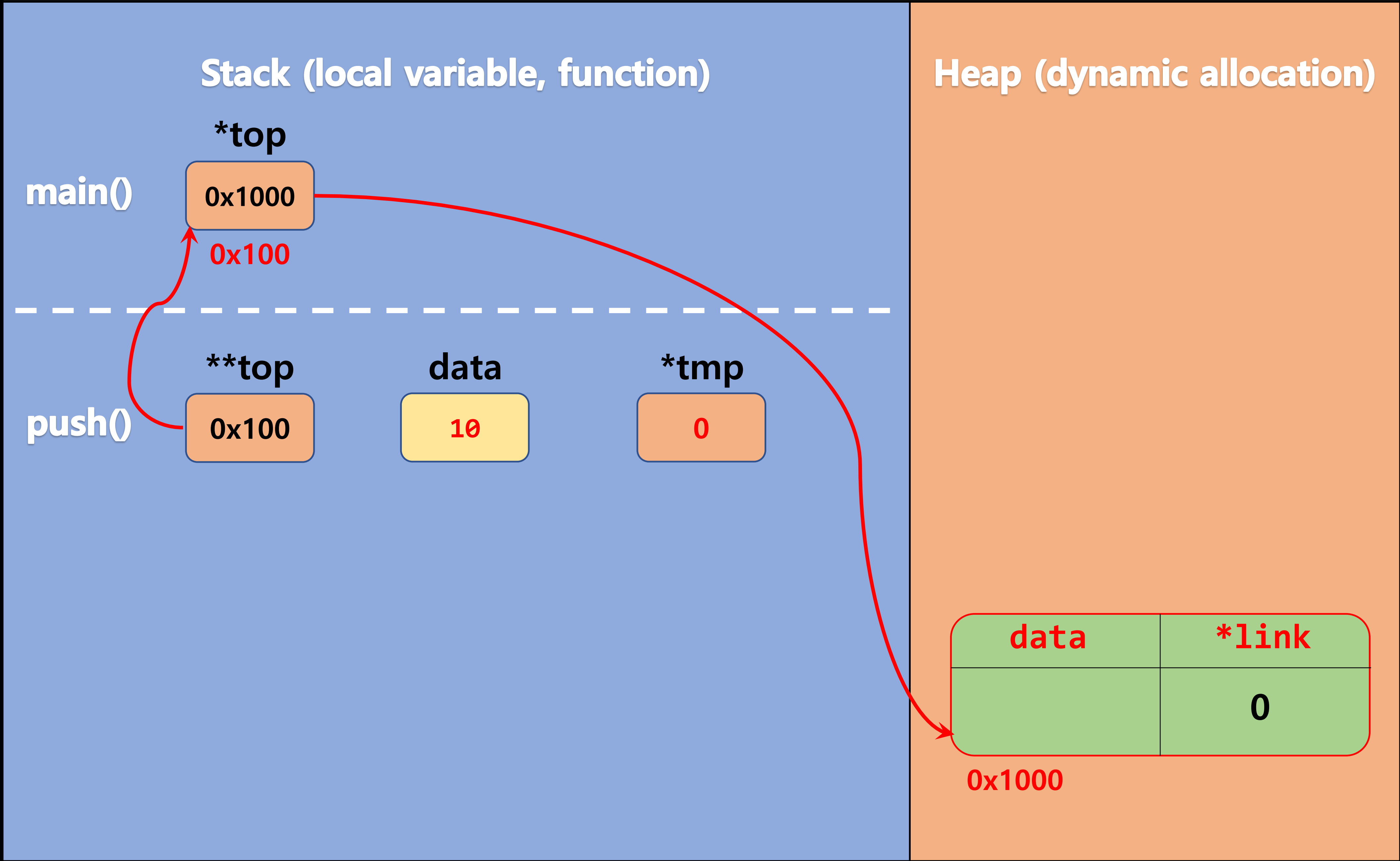


Data Struct

Stack

code)

```
(*top)->data = data;
```



Stack (local variable, function)

main()

*top

0x1000

0x100

push()

**top

0x100

data

10

*tmp

0

Heap (dynamic allocation)

data	*link
	0

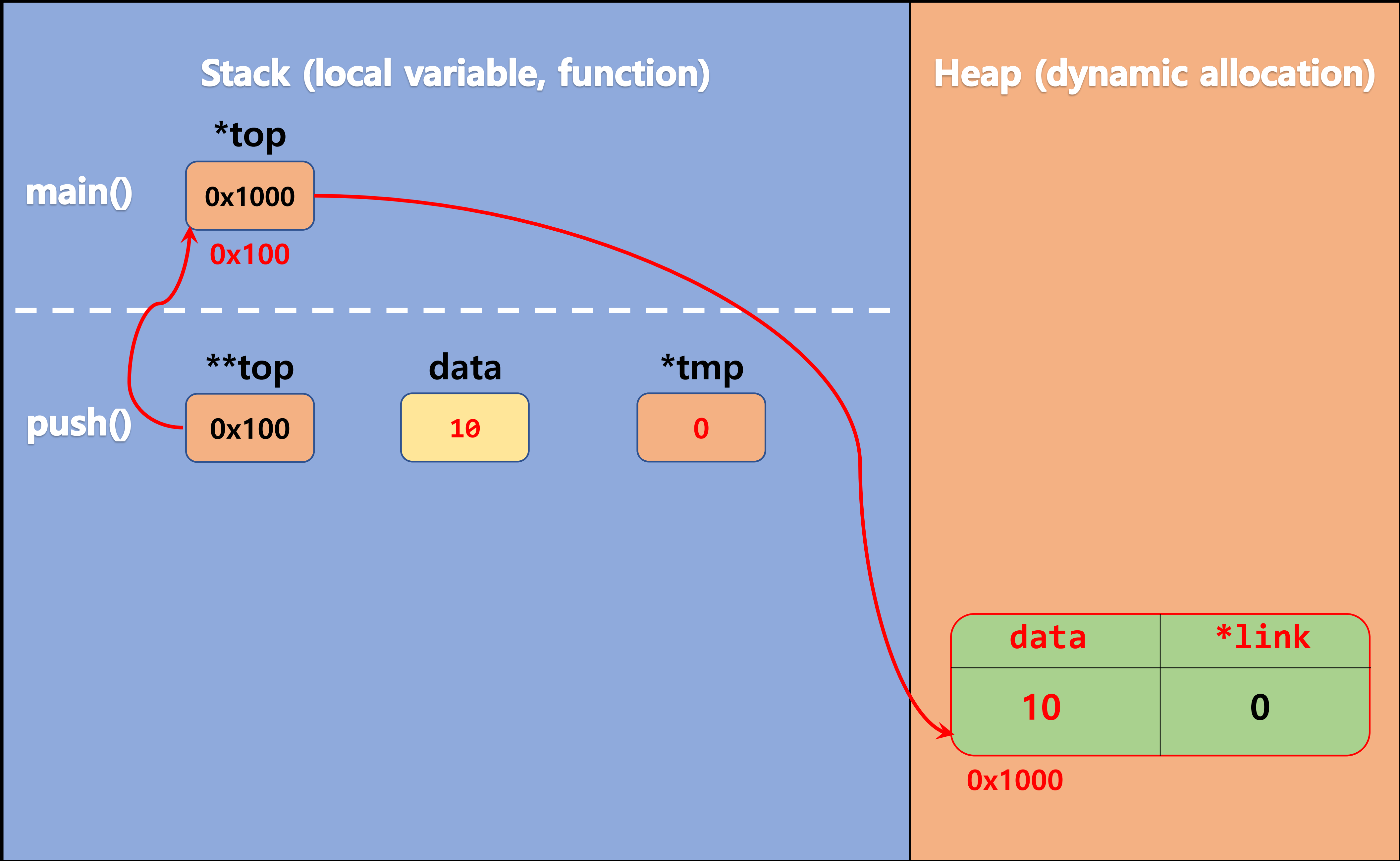
0x1000

Data Struct

Stack

code)

```
(*top)->data = data;
```

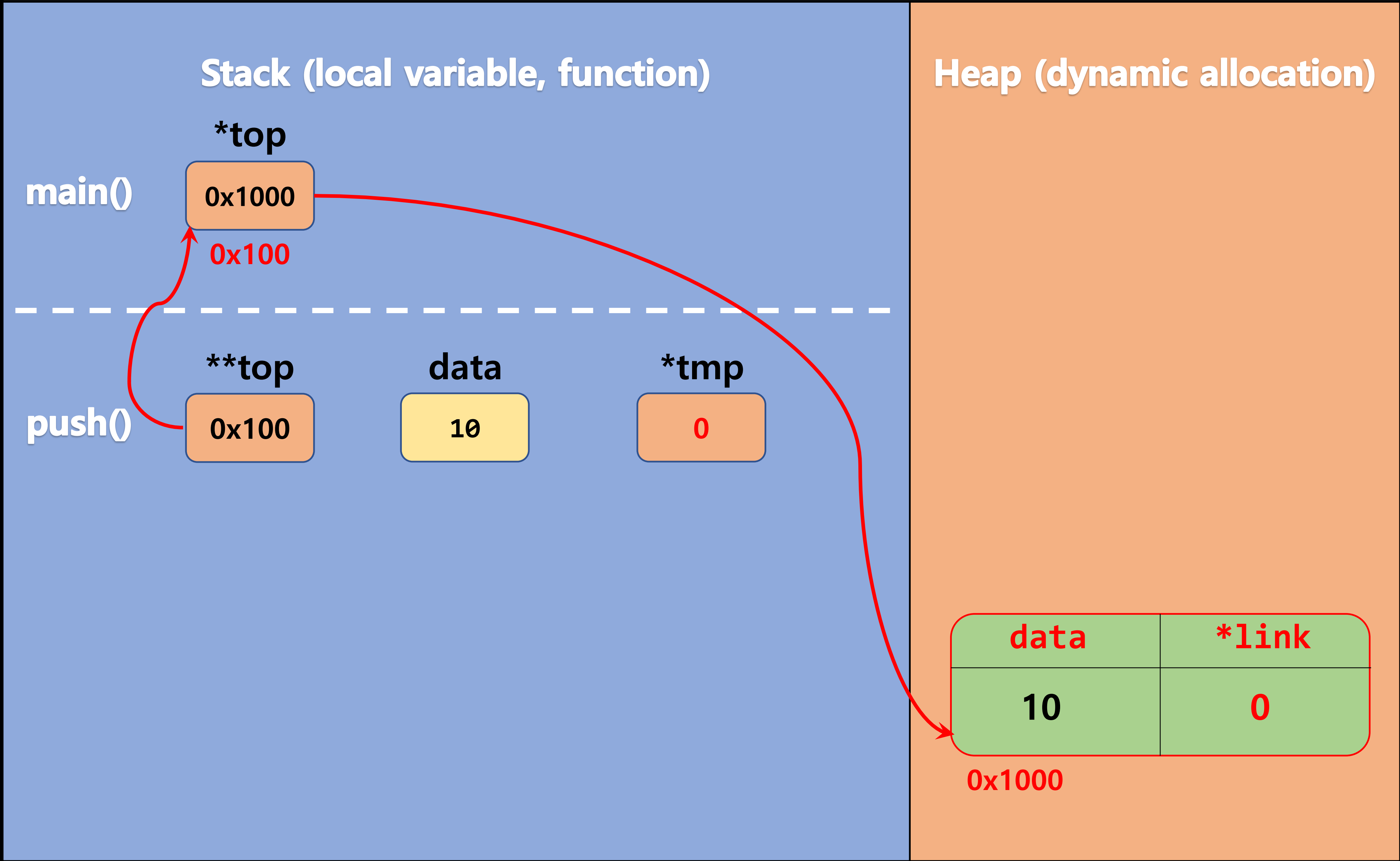


Data Struct

Stack

code)

```
(*top)->link = tmp;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x1000

0x100

push()

**top

0x100

data

10

*tmp

0

data

*link

10

0

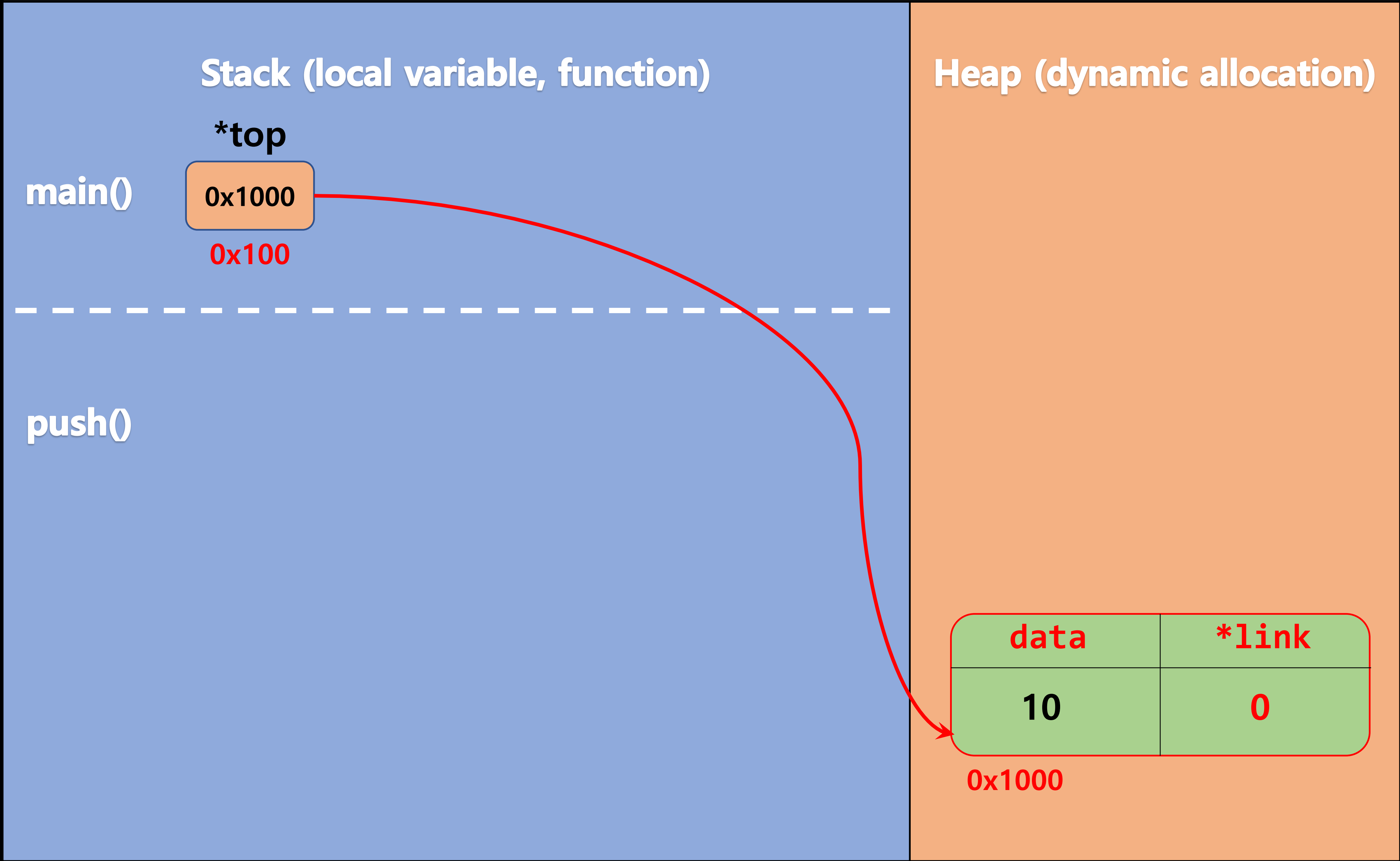
0x1000

Data Struct

Stack

code)

```
(*top)->link = tmp;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x1000

0x100

push()

data	*link
10	0

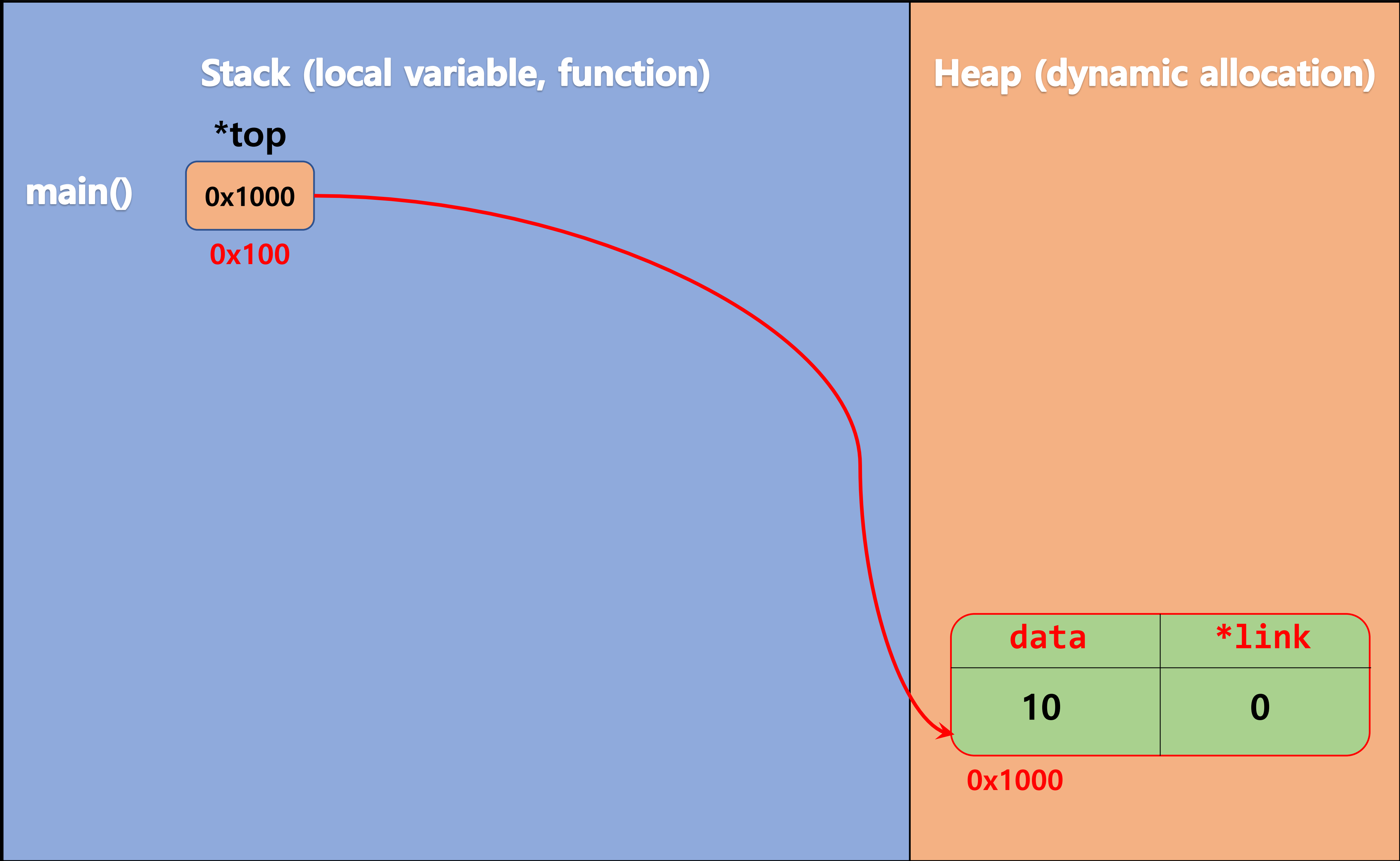
0x1000

Data Struct

Stack

code)

```
(*top)->link = tmp;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x1000

0x100

data	*link
10	0

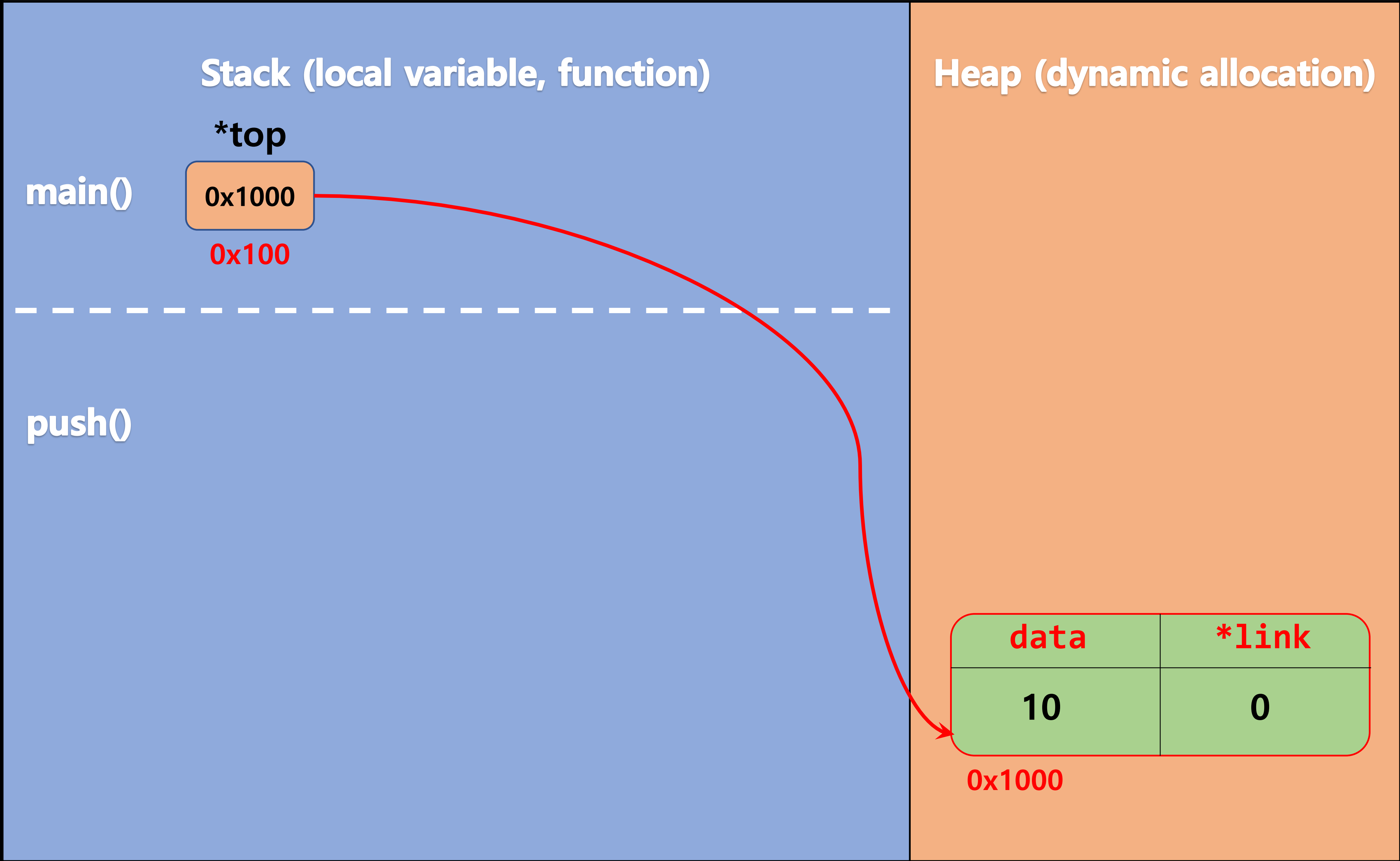
0x1000

Data Struct

Stack

code)

```
push(&top, 20);
```

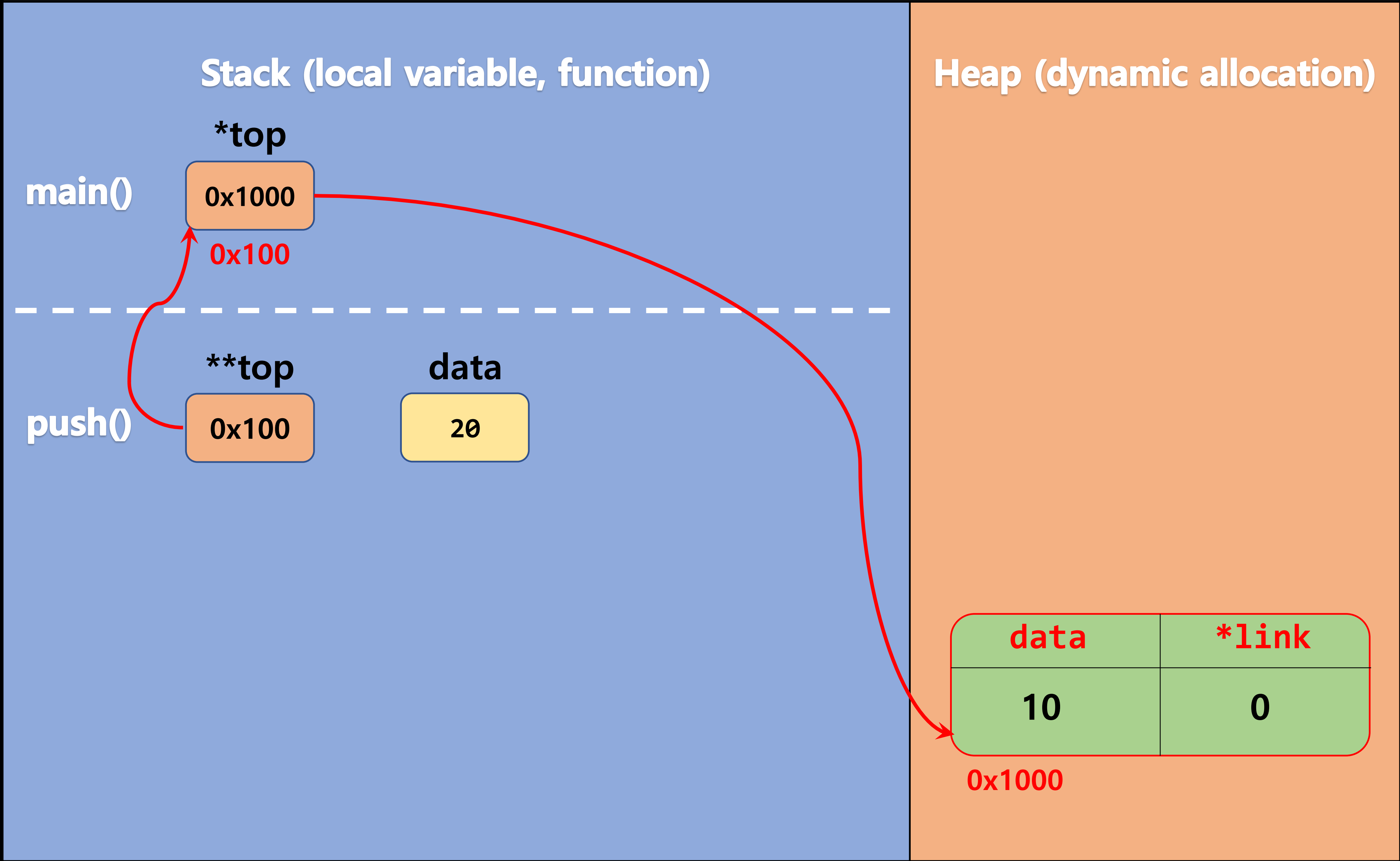


Data Struct

Stack

code)

```
push(&top, 20);
```

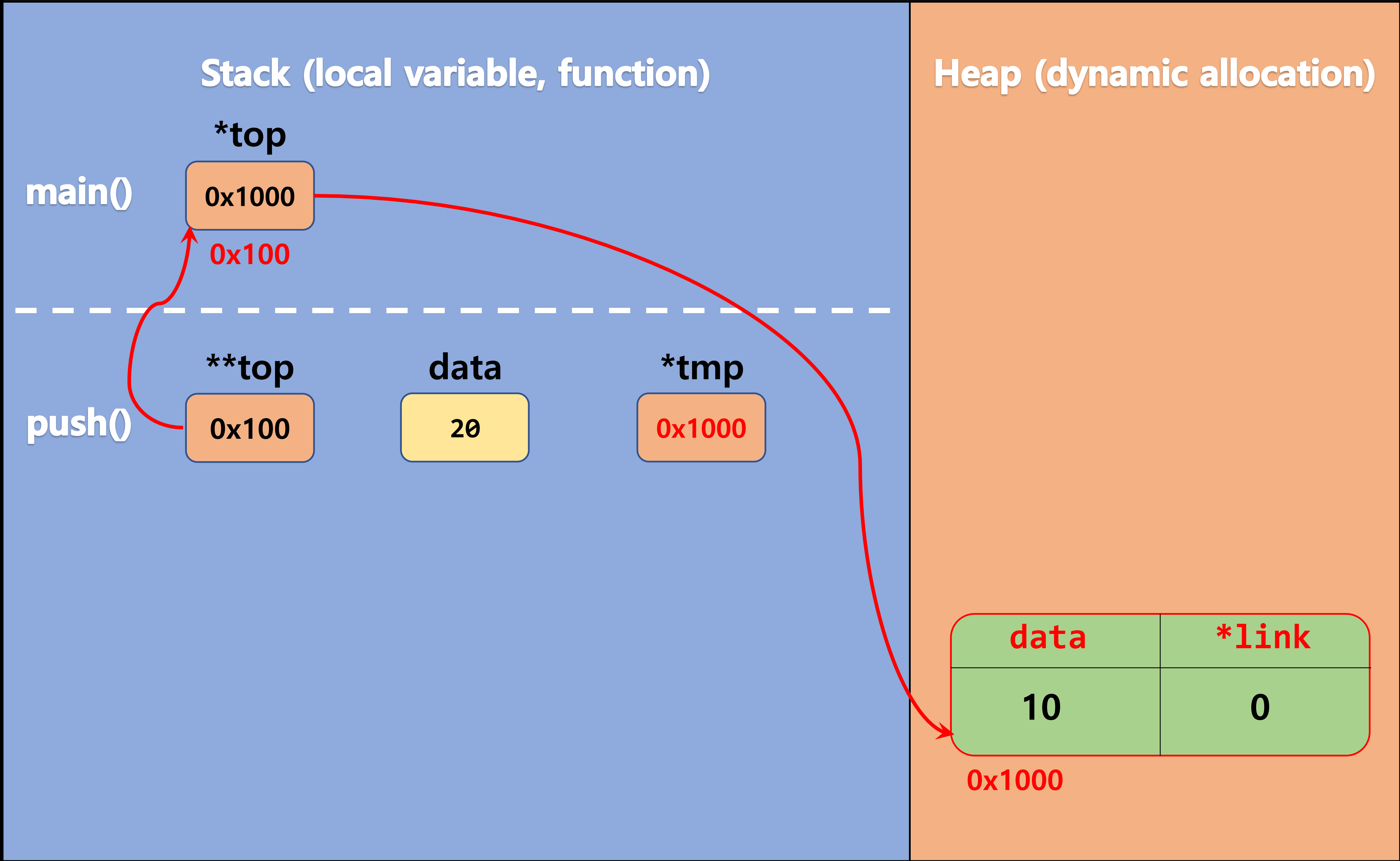


Data Struct

Stack

code)

```
Stack* tmp = *top;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x1000

0x100

push()

**top

0x100

data

20

*tmp

0x1000

data

*link

10

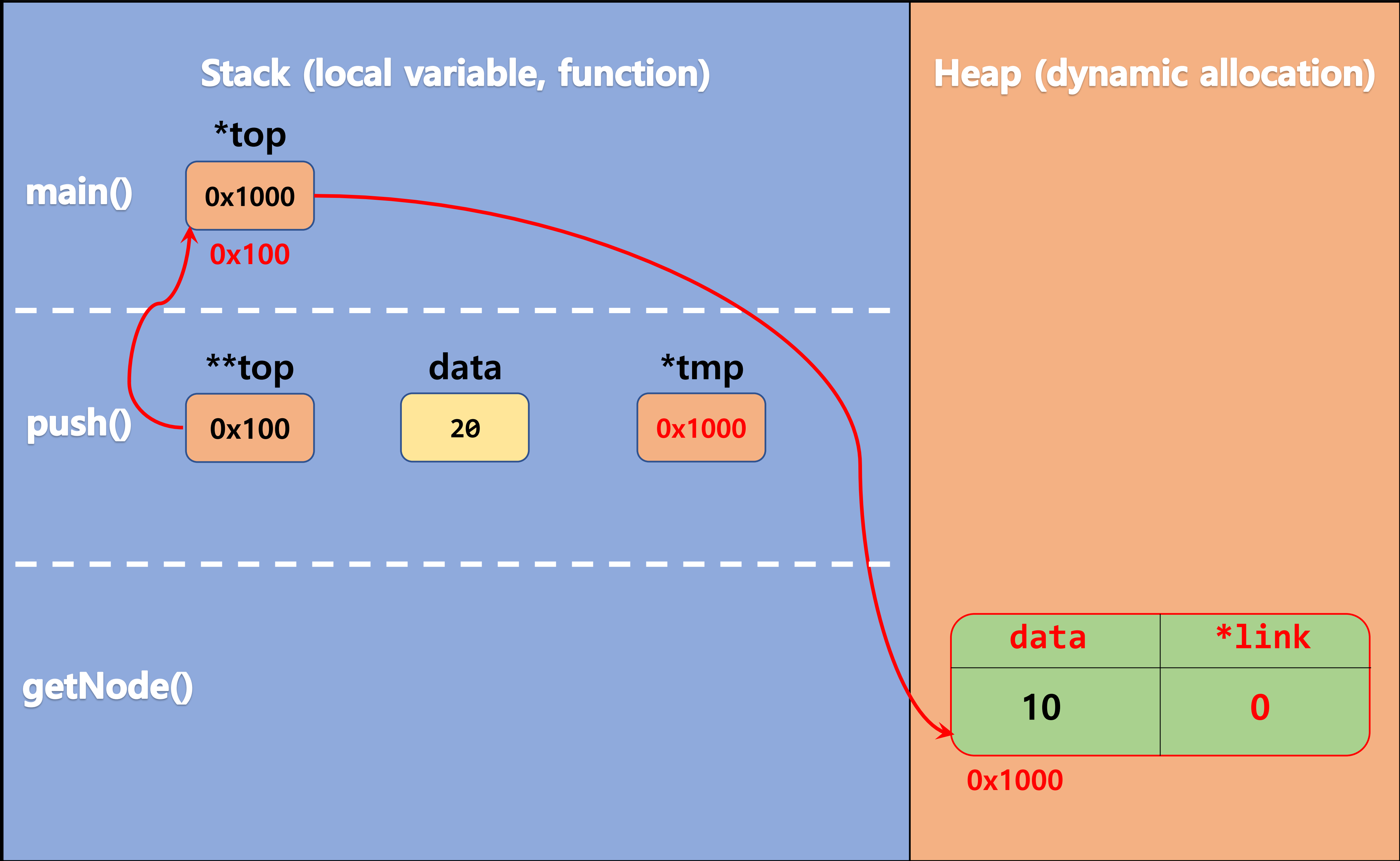
0

0x1000

Data Struct Stack

code)

```
*top = getNode();
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x1000

0x100

push()

**top

0x100

data

20

*tmp

0x1000

getNode()

data	*link
10	0

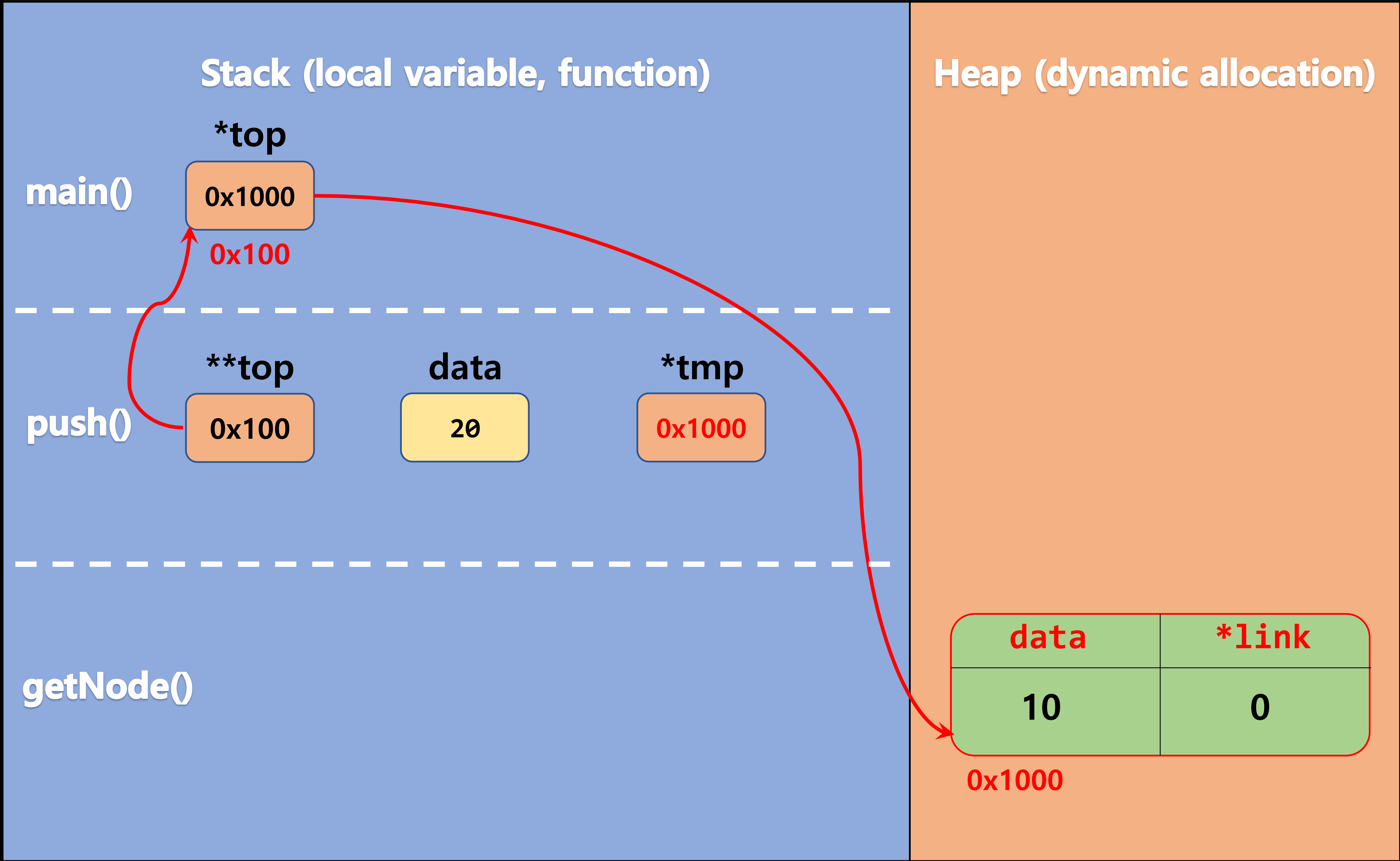
0x1000

Data Struct

Stack

code)

```c
Stack* newNode = (Stack*)malloc(sizeof(Stack));
```

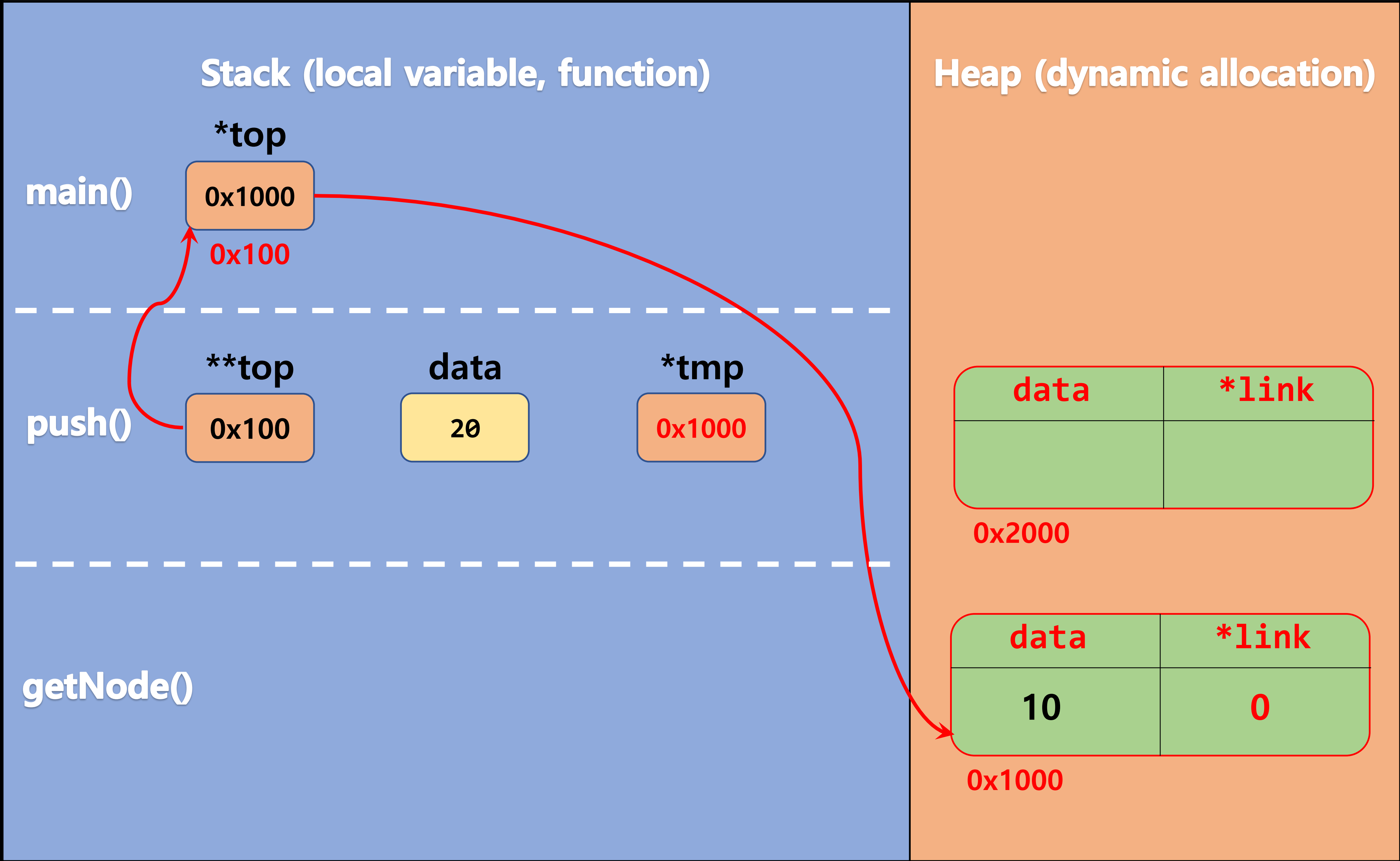


Data Struct

Stack

code)

```
Stack* newNode = (Stack*)malloc(sizeof(Stack));
```

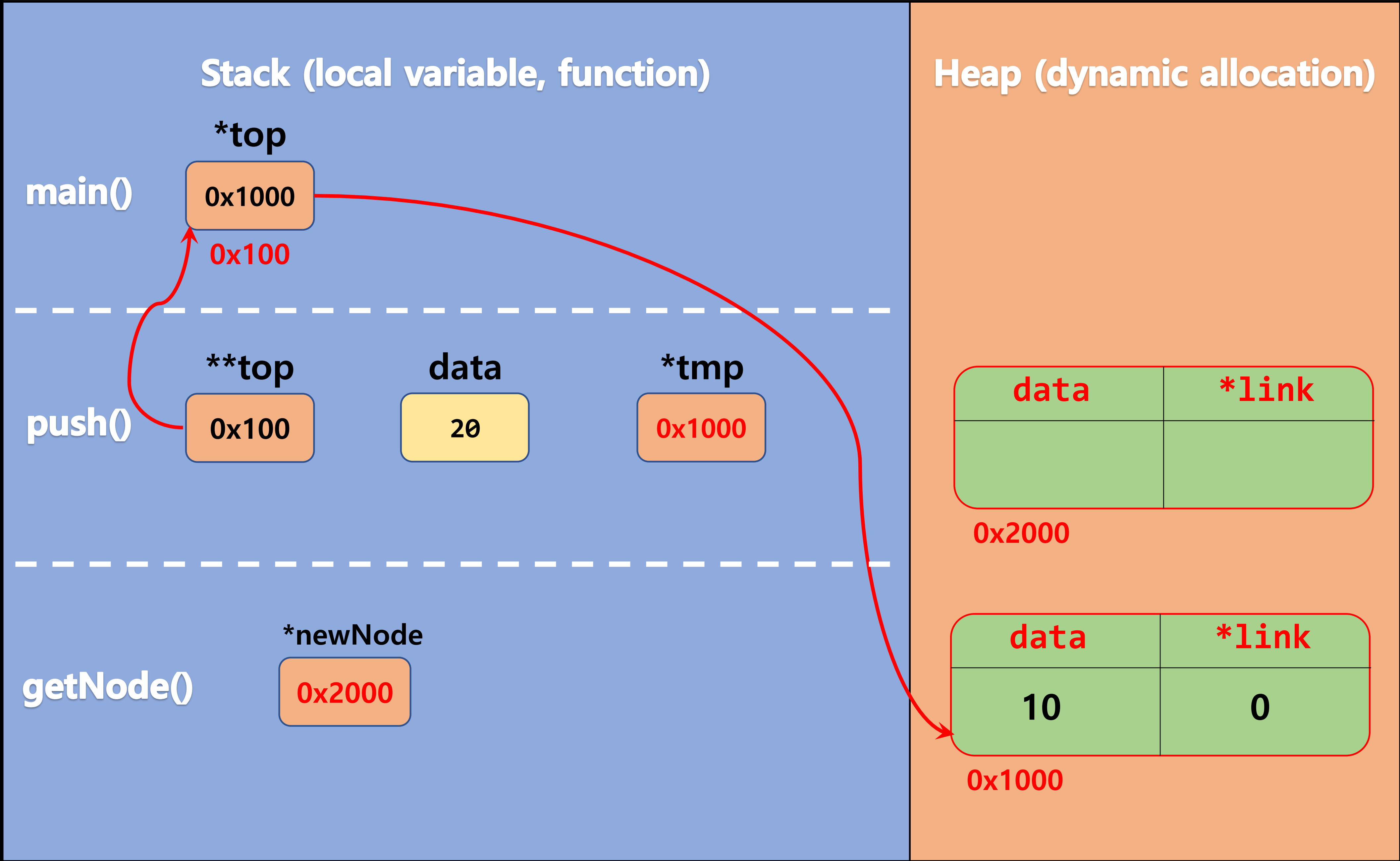


Data Struct

Stack

code)

```
Stack* newNode = (Stack*)malloc(sizeof(Stack));
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x1000

0x100

push()

**top

0x100

data

20

*tmp

0x1000

getNode()

*newNode

0x2000

data	*link

0x2000

data	*link
10	0

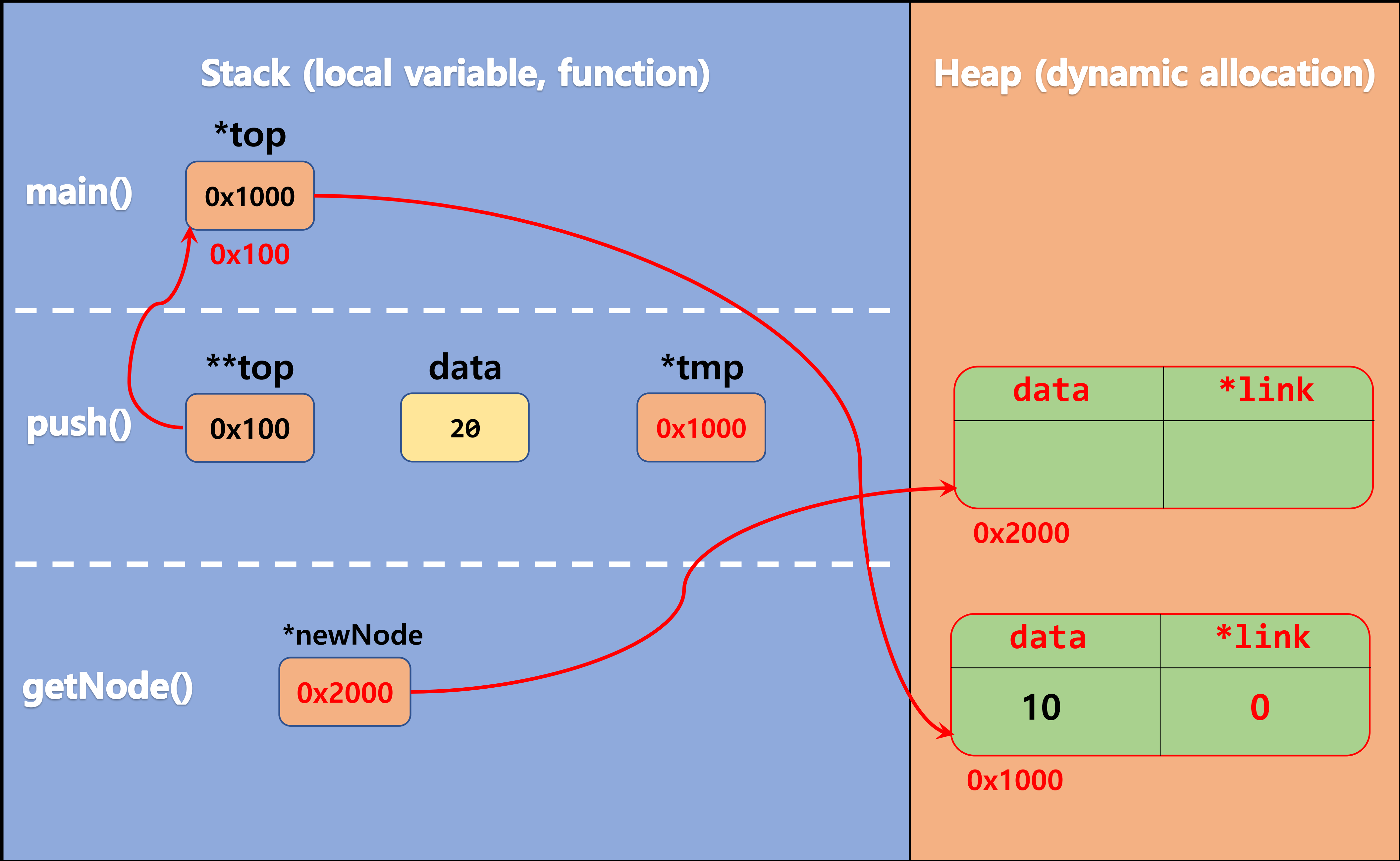
0x1000

Data Struct

Stack

code)

```
Stack* newNode = (Stack*)malloc(sizeof(Stack));
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x1000

0x100

push()

**top

0x100

data

20

*tmp

0x1000

getNode()

*newNode

0x2000

data	*link

0x2000

data	*link
10	0

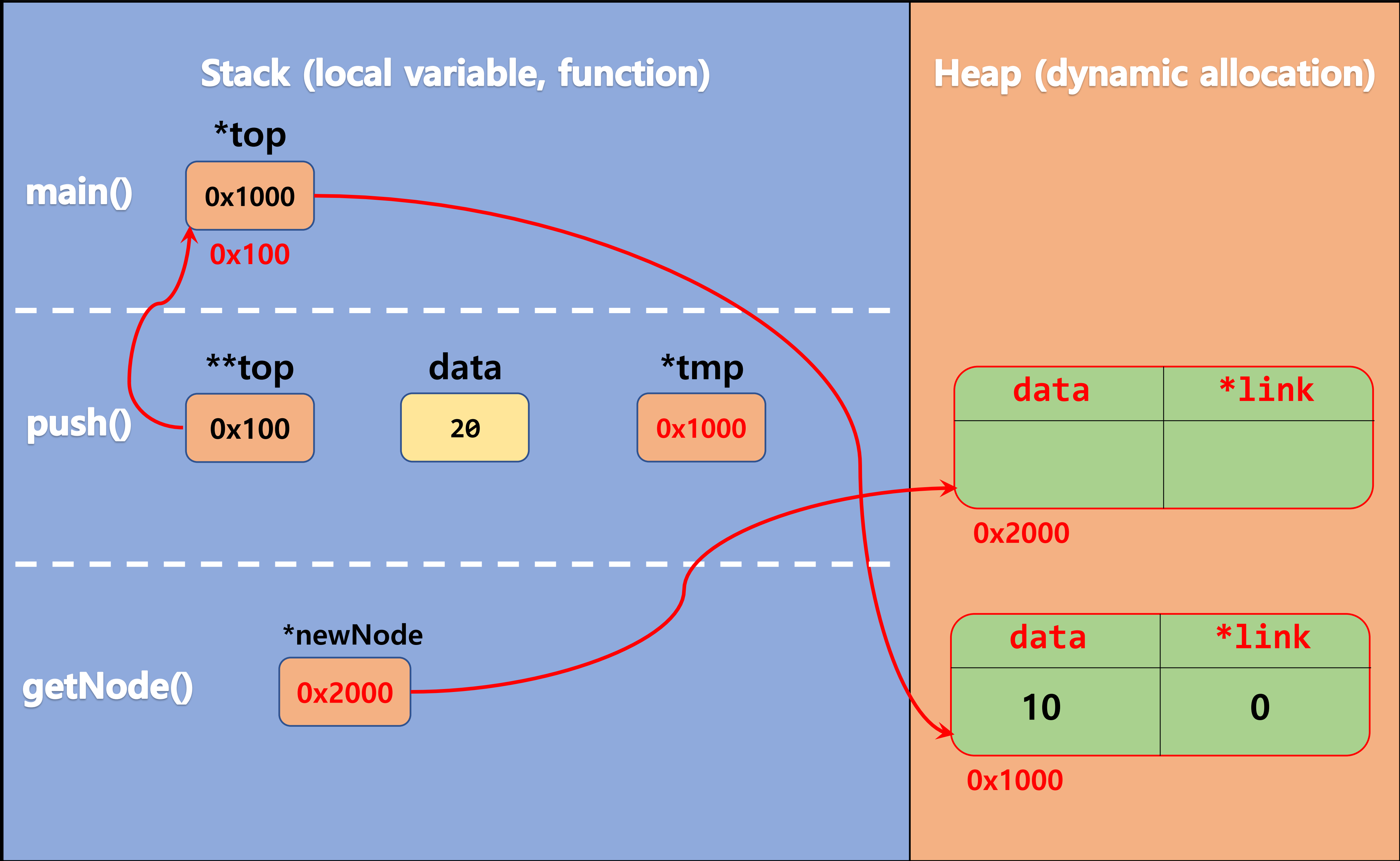
0x1000

Data Struct

Stack

code)

```
newNode->link = NULL;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x1000

0x100

push()

**top

0x100

data

20

*tmp

0x1000

getNode()

*newNode

0x2000

data	*link

0x2000

data	*link
10	0

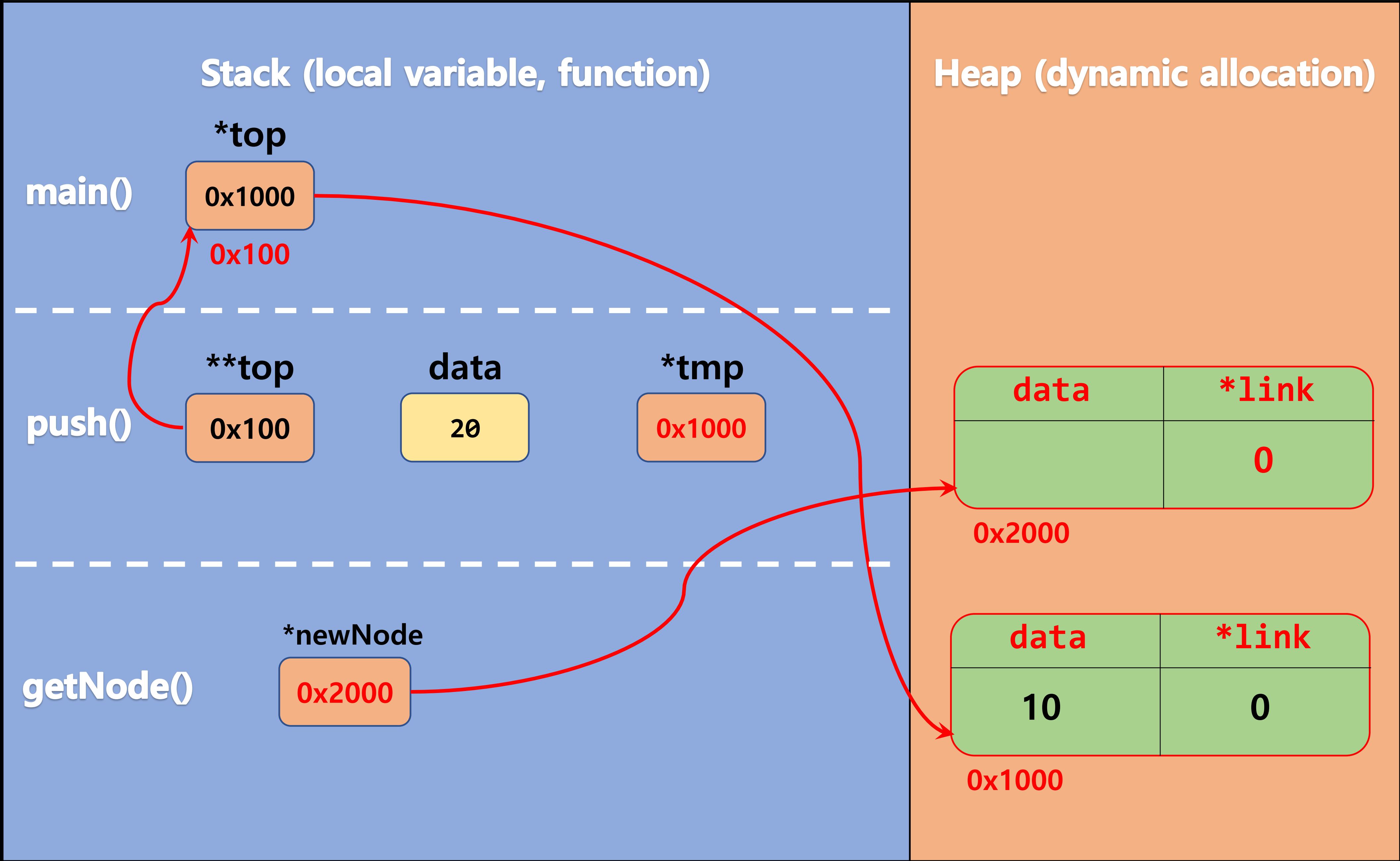
0x1000

Data Struct

Stack

code)

```
newNode->link = NULL;
```

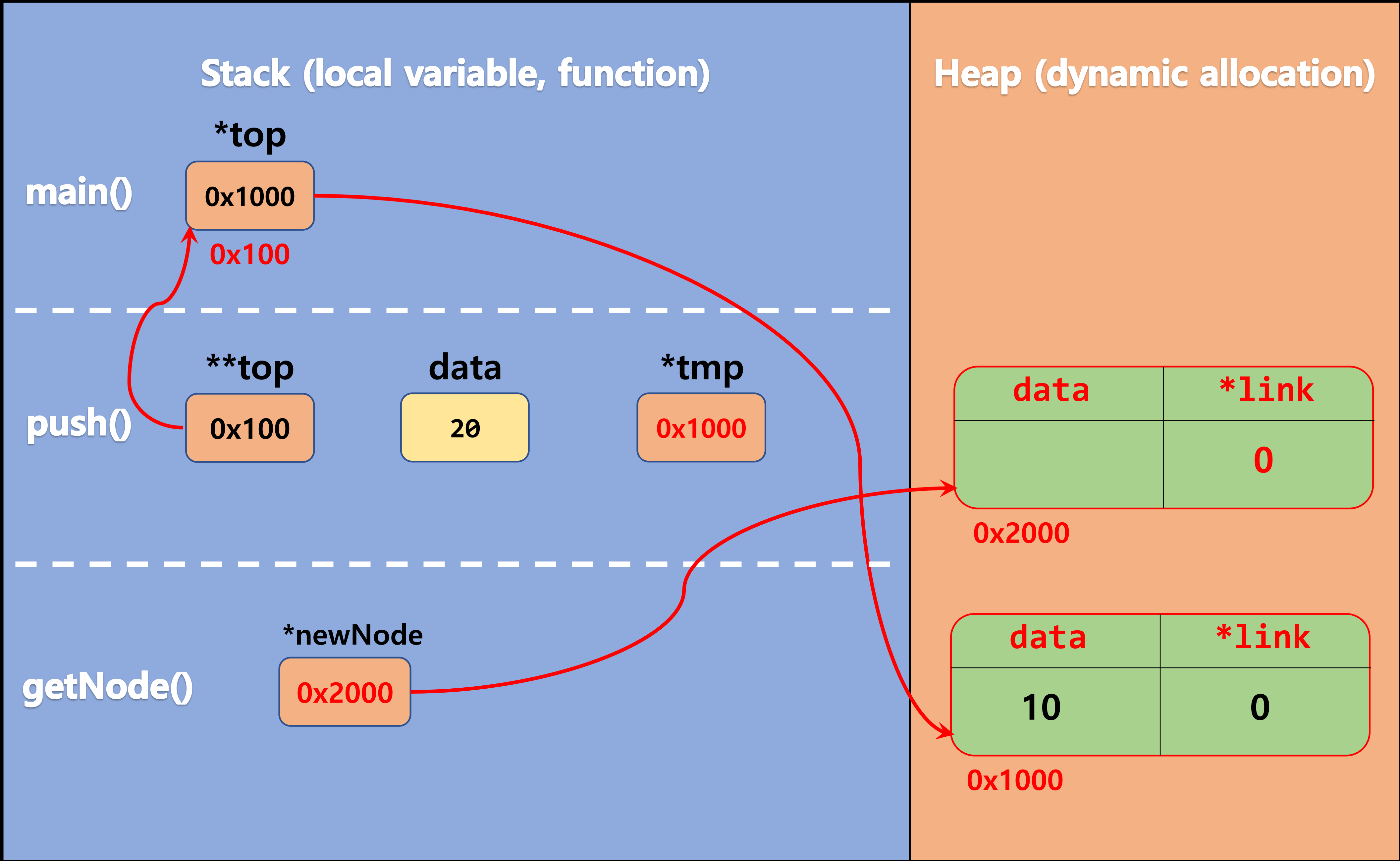


Data Struct

Stack

code)

```
return newNode;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x1000

0x100

push()

**top

0x100

data

20

*tmp

0x1000

getNode()

*newNode

0x2000

data	*link
	0

0x2000

data	*link
10	0

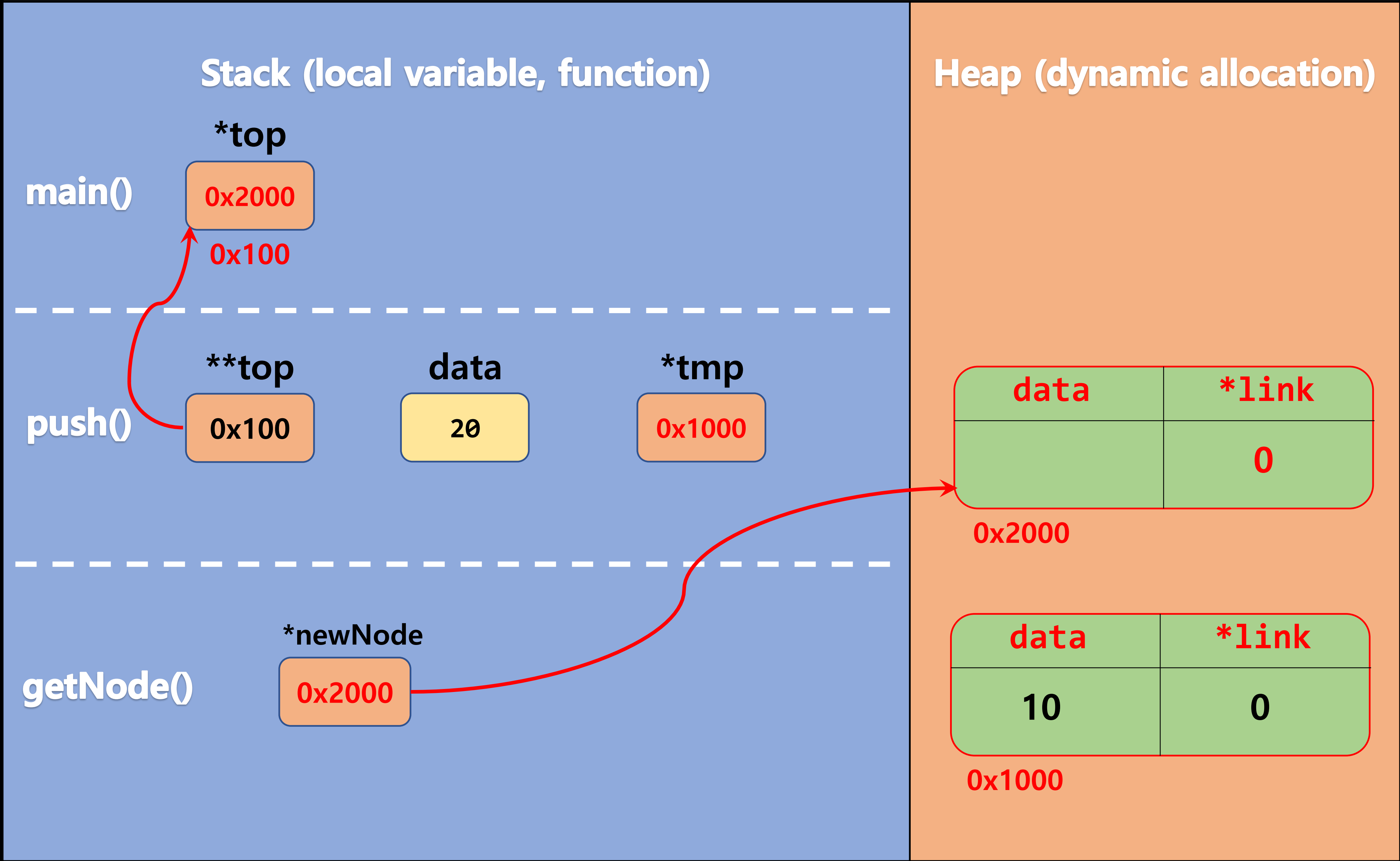
0x1000

Data Struct

Stack

code)

```
return newNode;
```

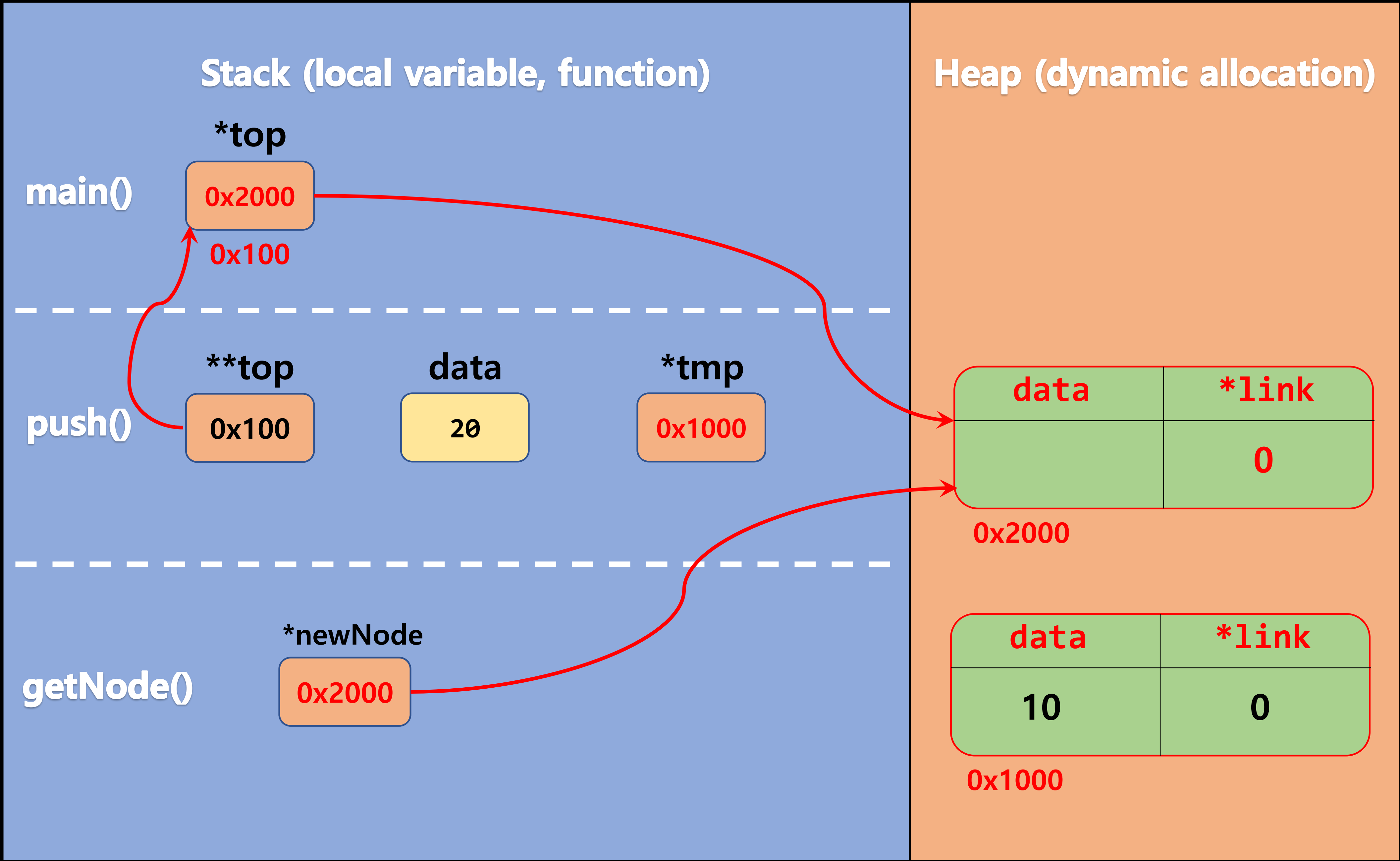


Data Struct

Stack

code)

```
return newNode;
```

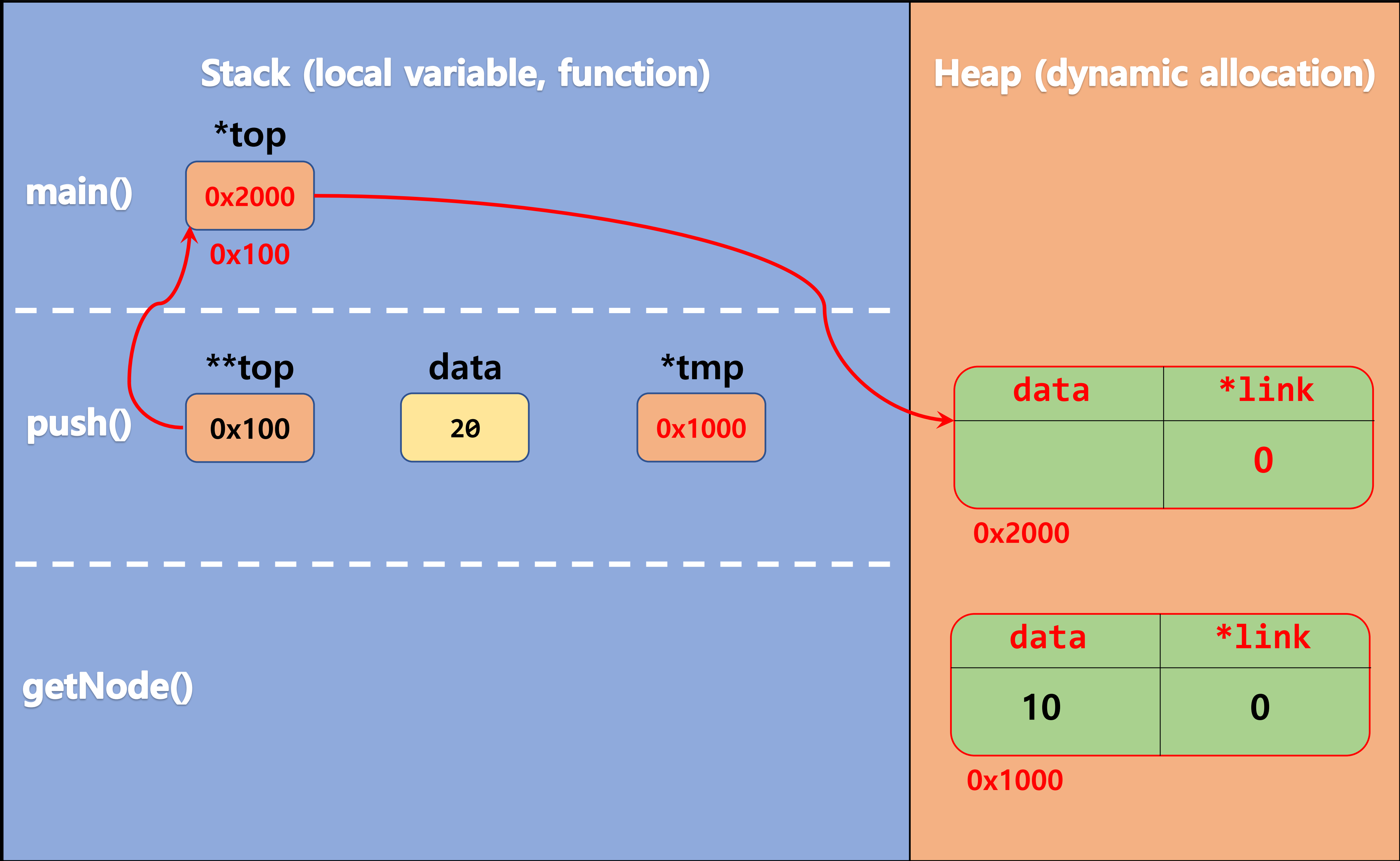


Data Struct

Stack

code)

```c
return newNode;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x2000

0x100

push()

**top

0x100

data

20

*tmp

0x1000

getNode()

data	*link
	0

0x2000

data	*link
10	0

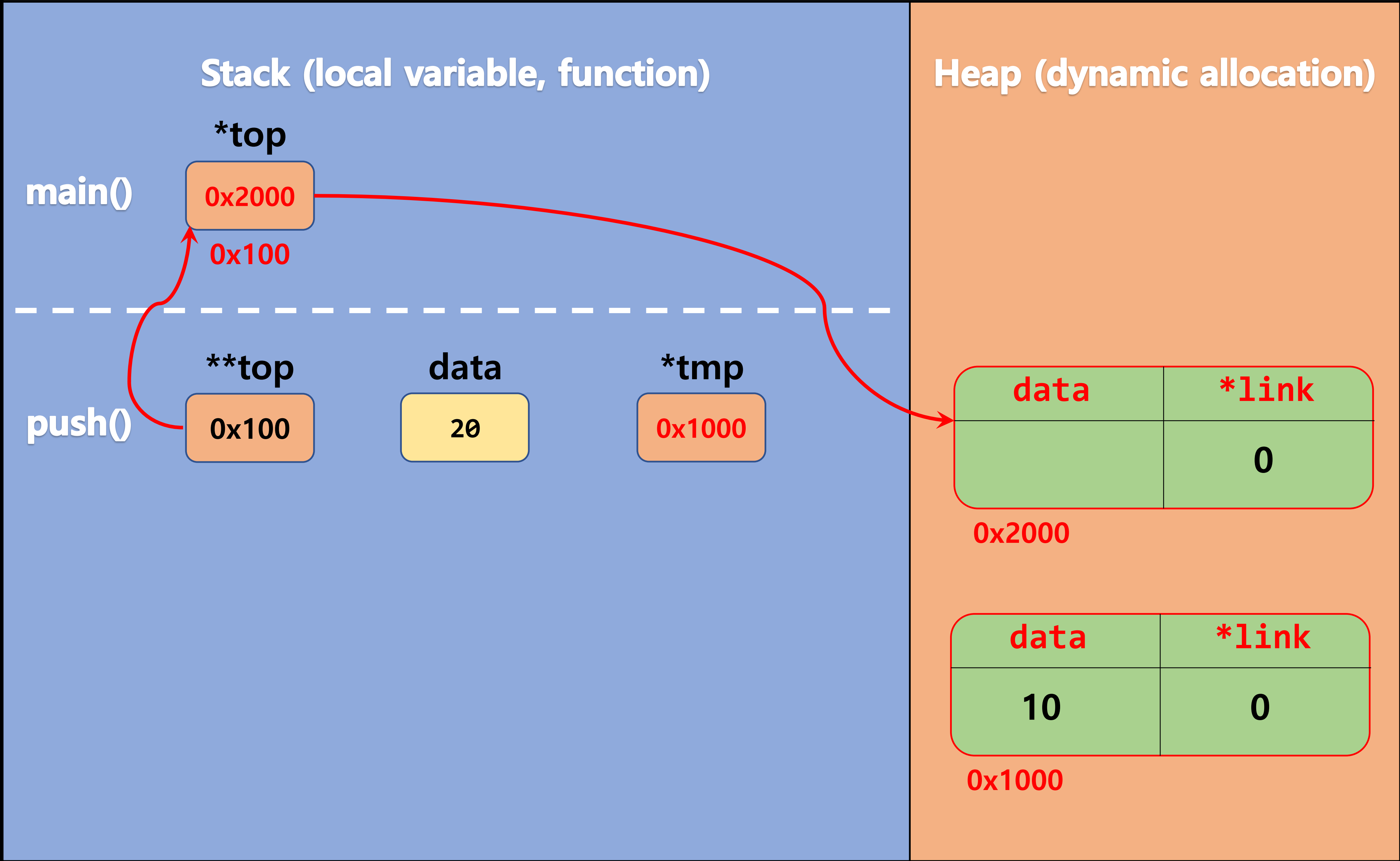
0x1000

Data Struct

Stack

code)

```
return newNode;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x2000

0x100

push()

**top

0x100

data

20

*tmp

0x1000

data	*link
	0

0x2000

data	*link
10	0

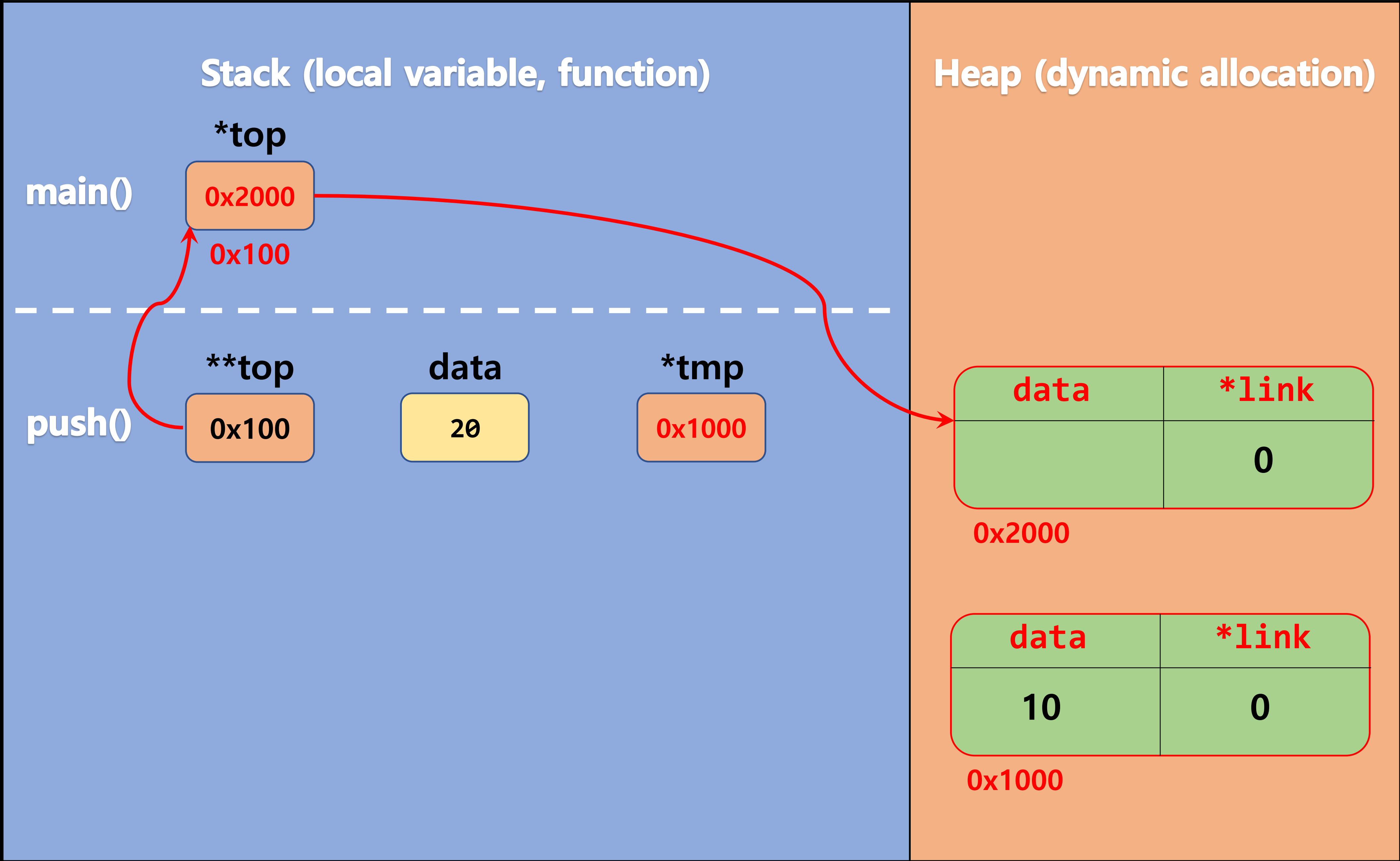
0x1000

Data Struct

Stack

code)

```
(*top)->data = data;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x2000

0x100

push()

**top

0x100

data

20

*tmp

0x1000

data	*link
	0

0x2000

data	*link
10	0

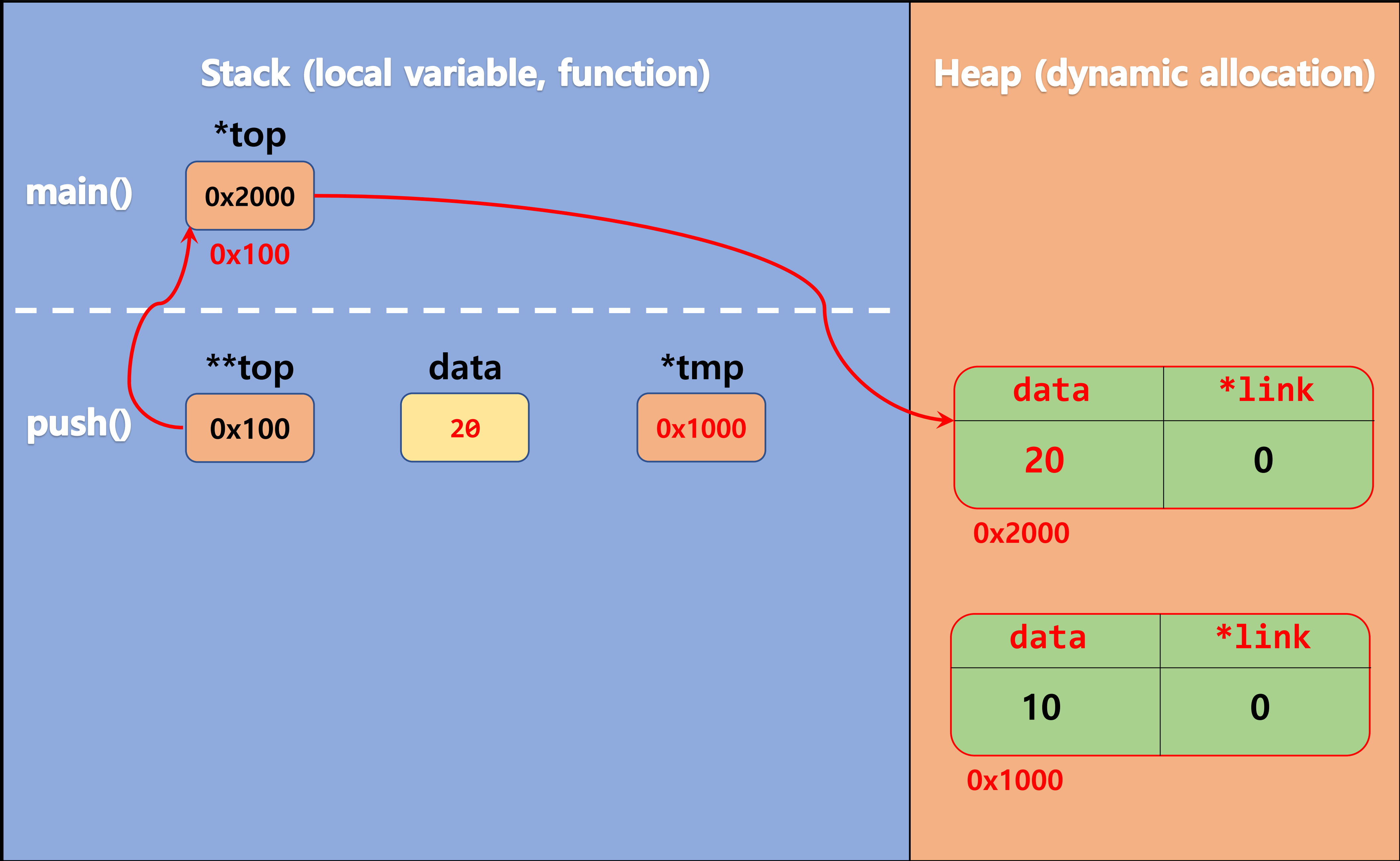
0x1000

Data Struct

Stack

code)

```
(*top)->data = data;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x2000

0x100

push()

**top

0x100

data

20

*tmp

0x1000

data	*link
20	0

0x2000

data	*link
10	0

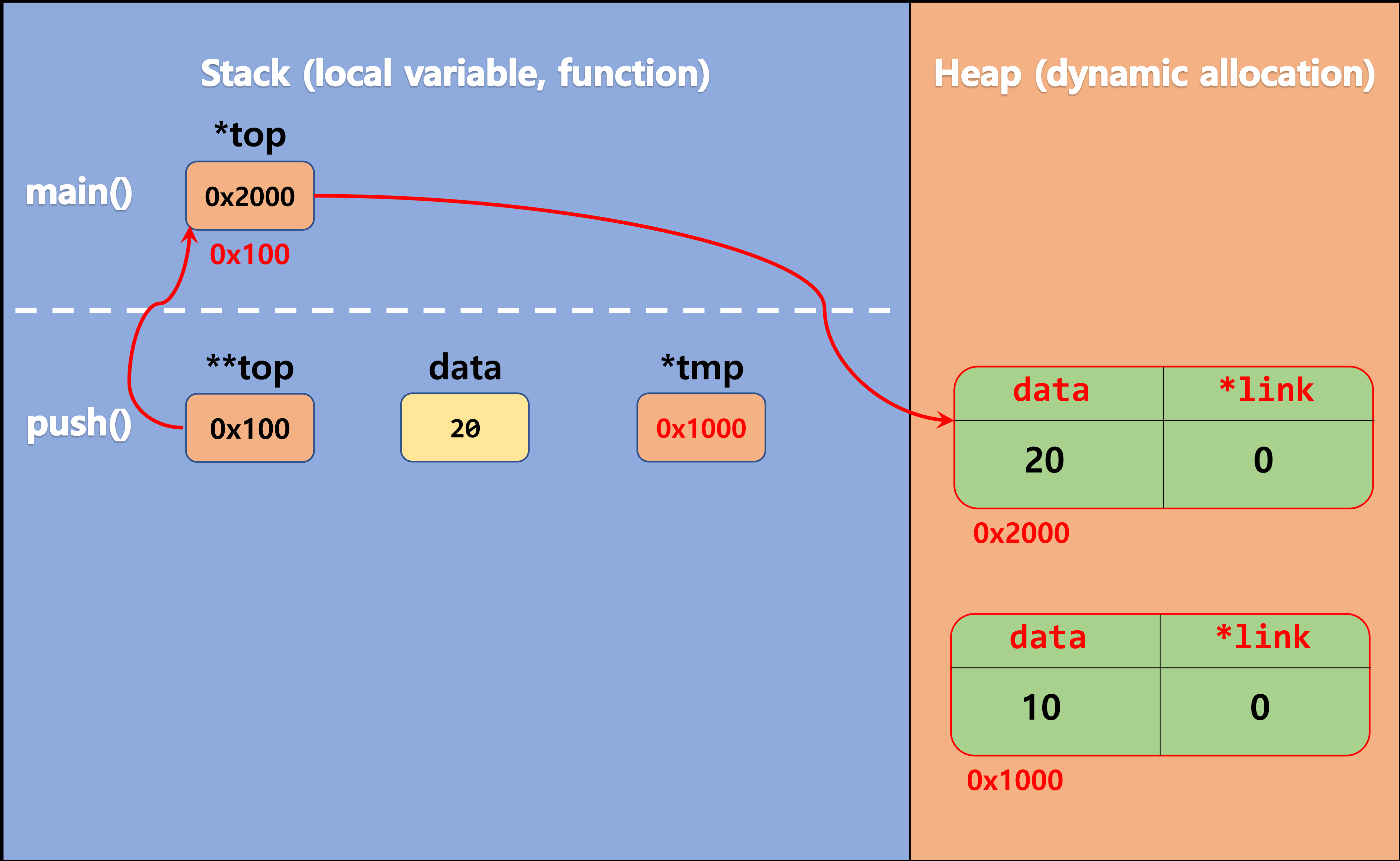
0x1000

Data Struct

Stack

code)

```
(*top)->link = tmp;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x2000

0x100

push()

**top

0x100

data

20

*tmp

0x1000

data

*link

20

0

0x2000

data

*link

10

0

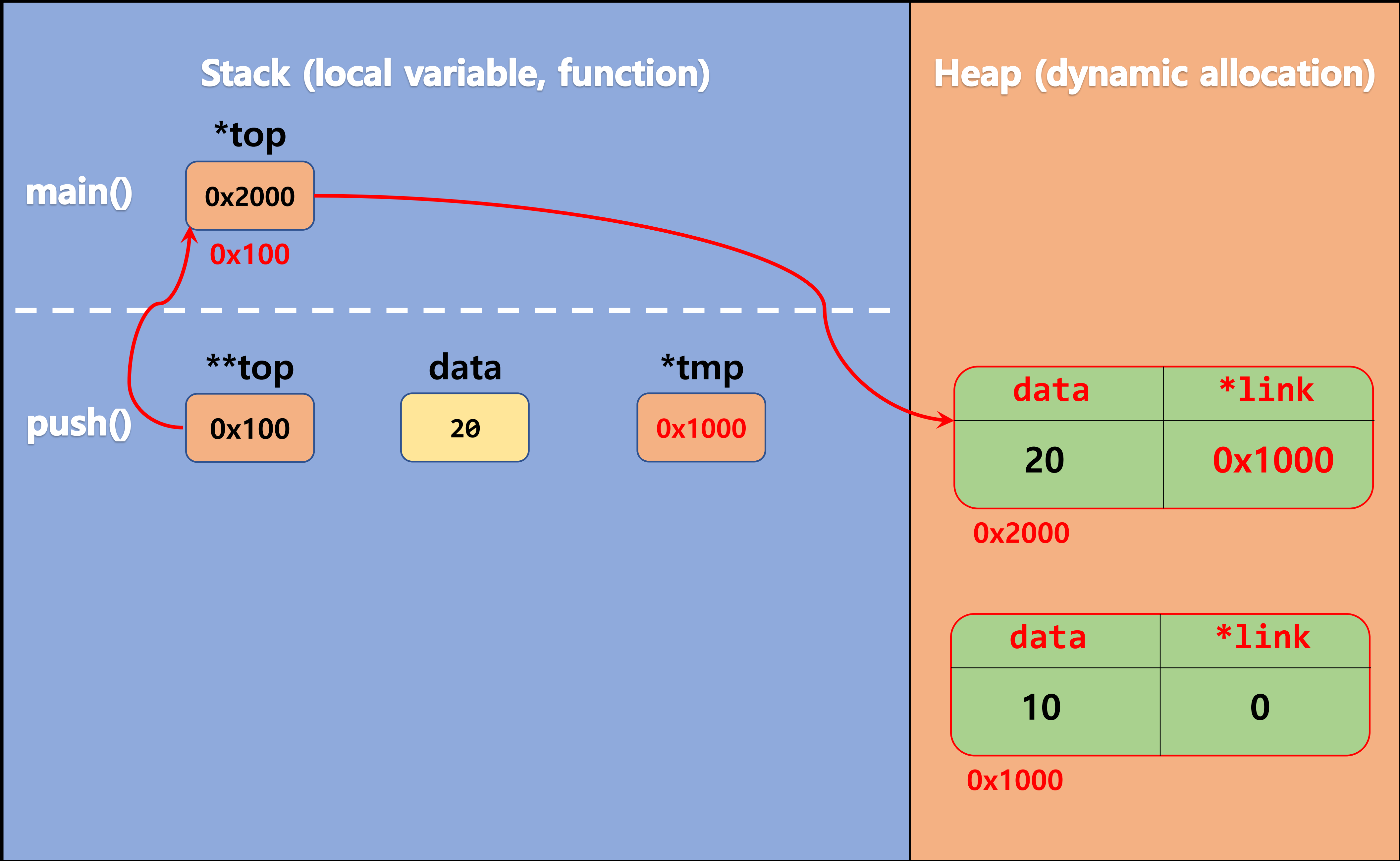
0x1000

Data Struct

Stack

code)

```
(*top)->link = tmp;
```

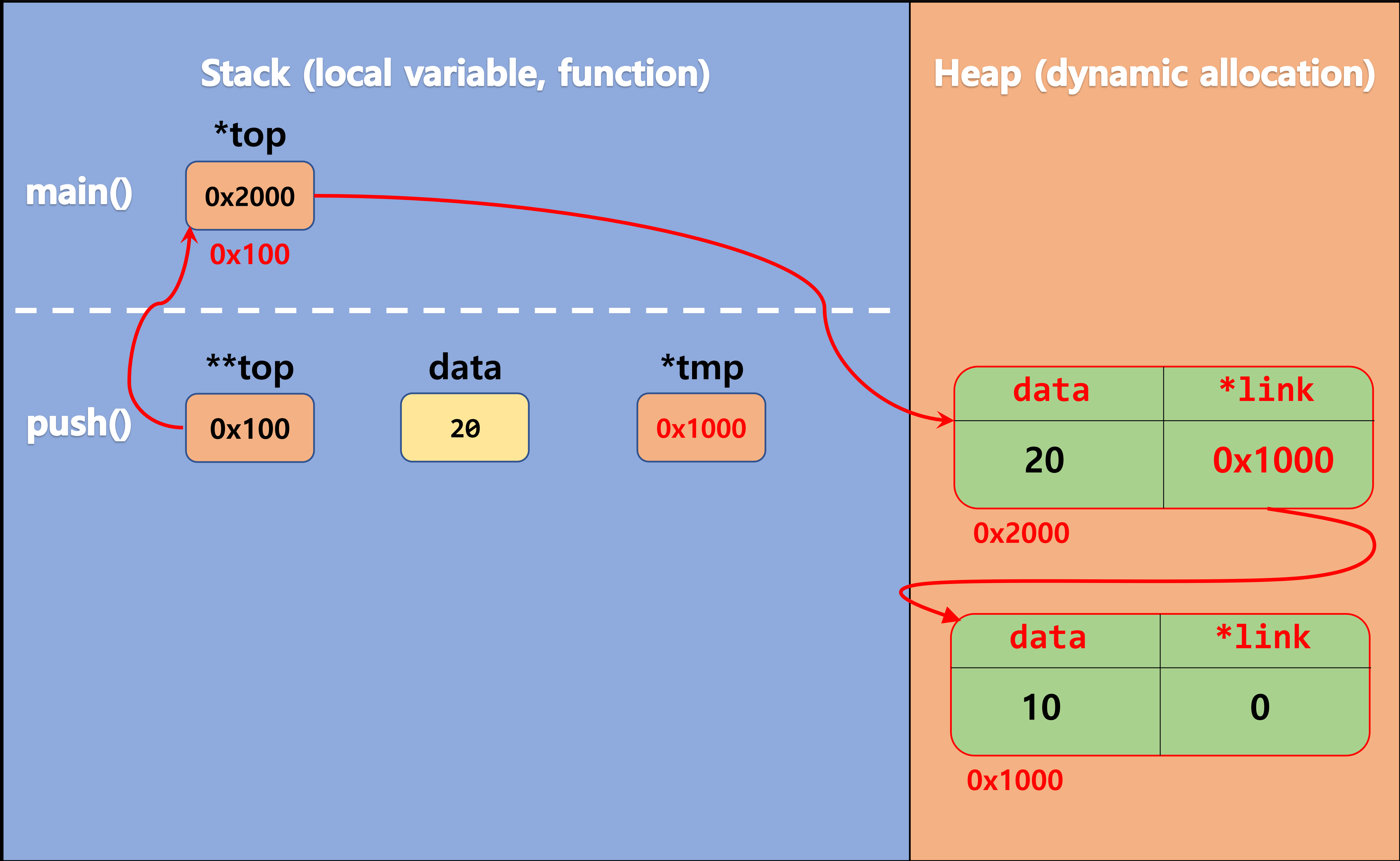


Data Struct

Stack

code)

```
(*top)->link = tmp;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x2000

0x100

push()

**top

0x100

data

20

*tmp

0x1000

data	*link
20	0x1000

0x2000

data	*link
10	0

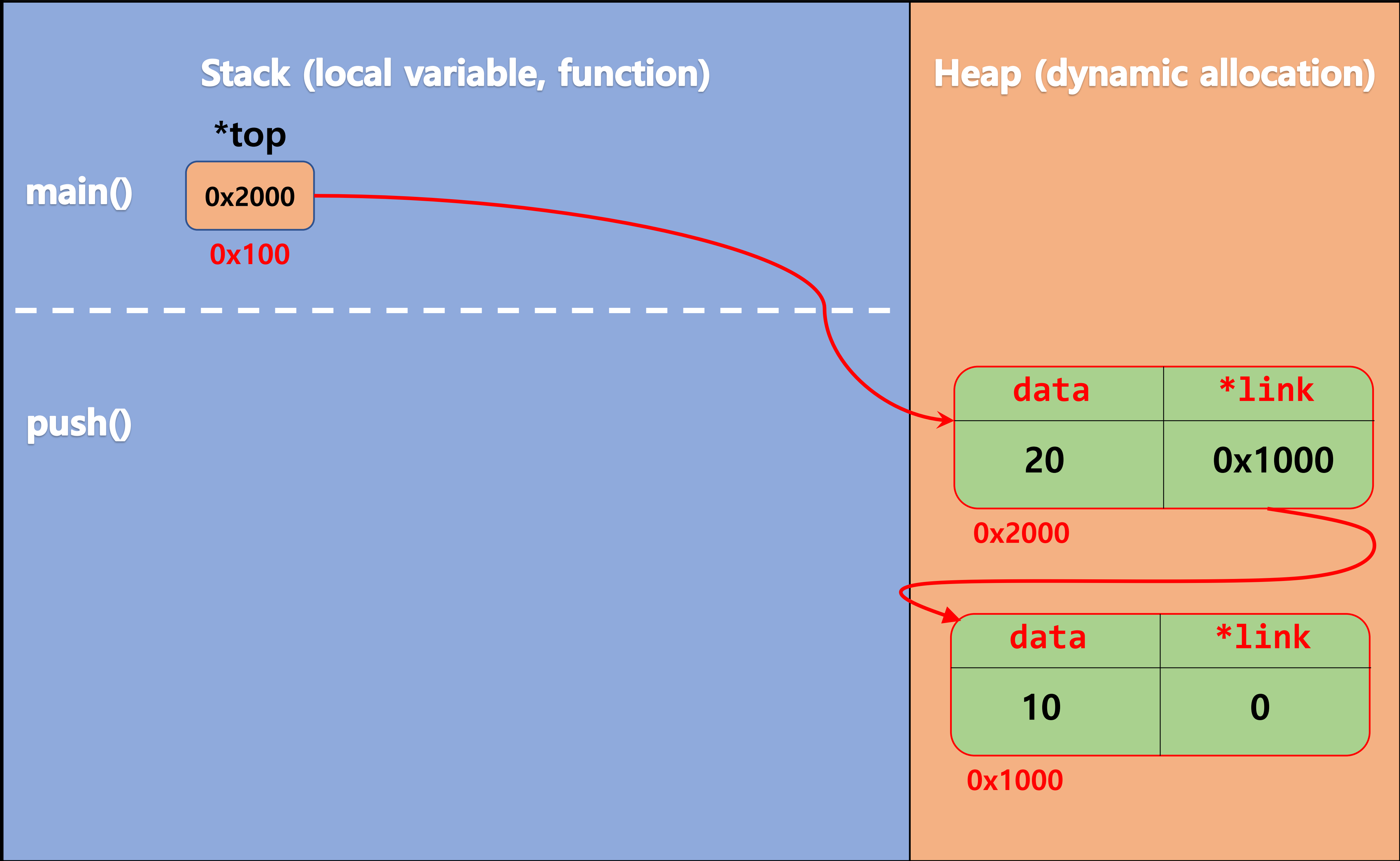
0x1000

Data Struct

Stack

code)

```
(*top)->link = tmp;
```



Stack (local variable, function)

main()

*top

0x2000

0x100

push()

Heap (dynamic allocation)

data	*link
20	0x1000

0x2000

data	*link
10	0

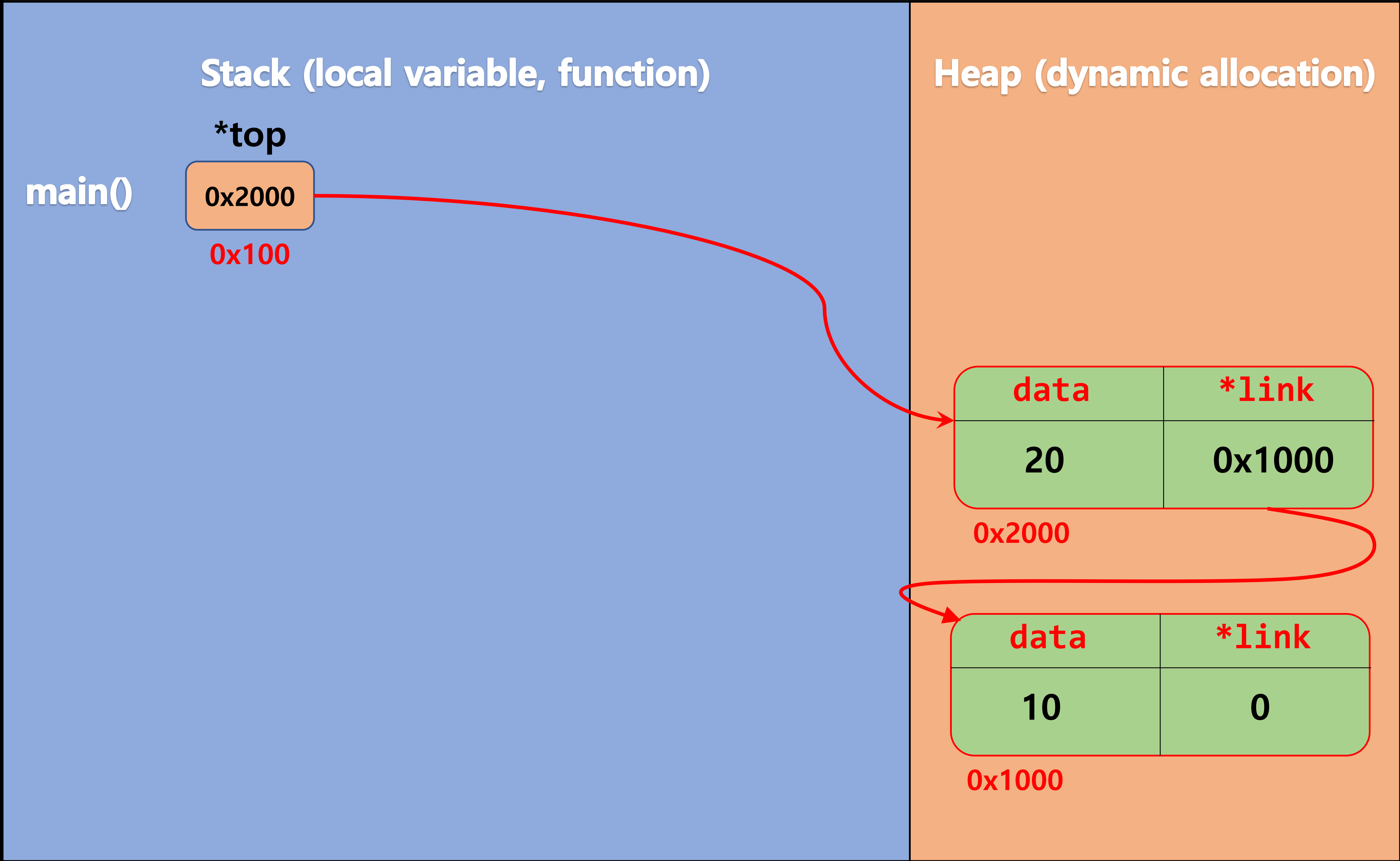
0x1000

Data Struct

Stack

code)

```
(*top)->link = tmp;
```

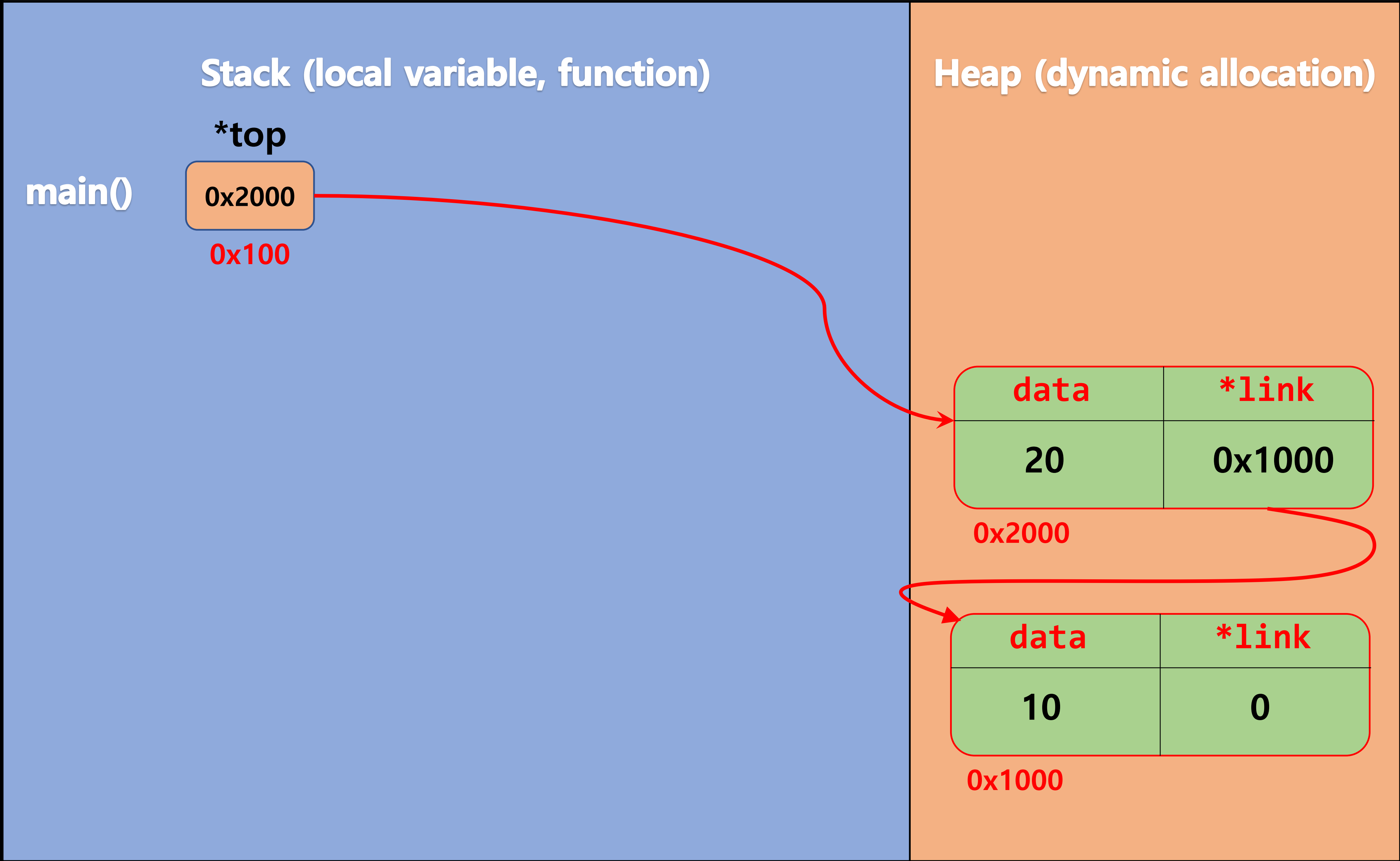


Data Struct

Stack

code)

```
push(&top, 30);
```



Stack (local variable, function)

main()

*top

0x2000

0x100

Heap (dynamic allocation)

data	*link
20	0x1000

0x2000

data	*link
10	0

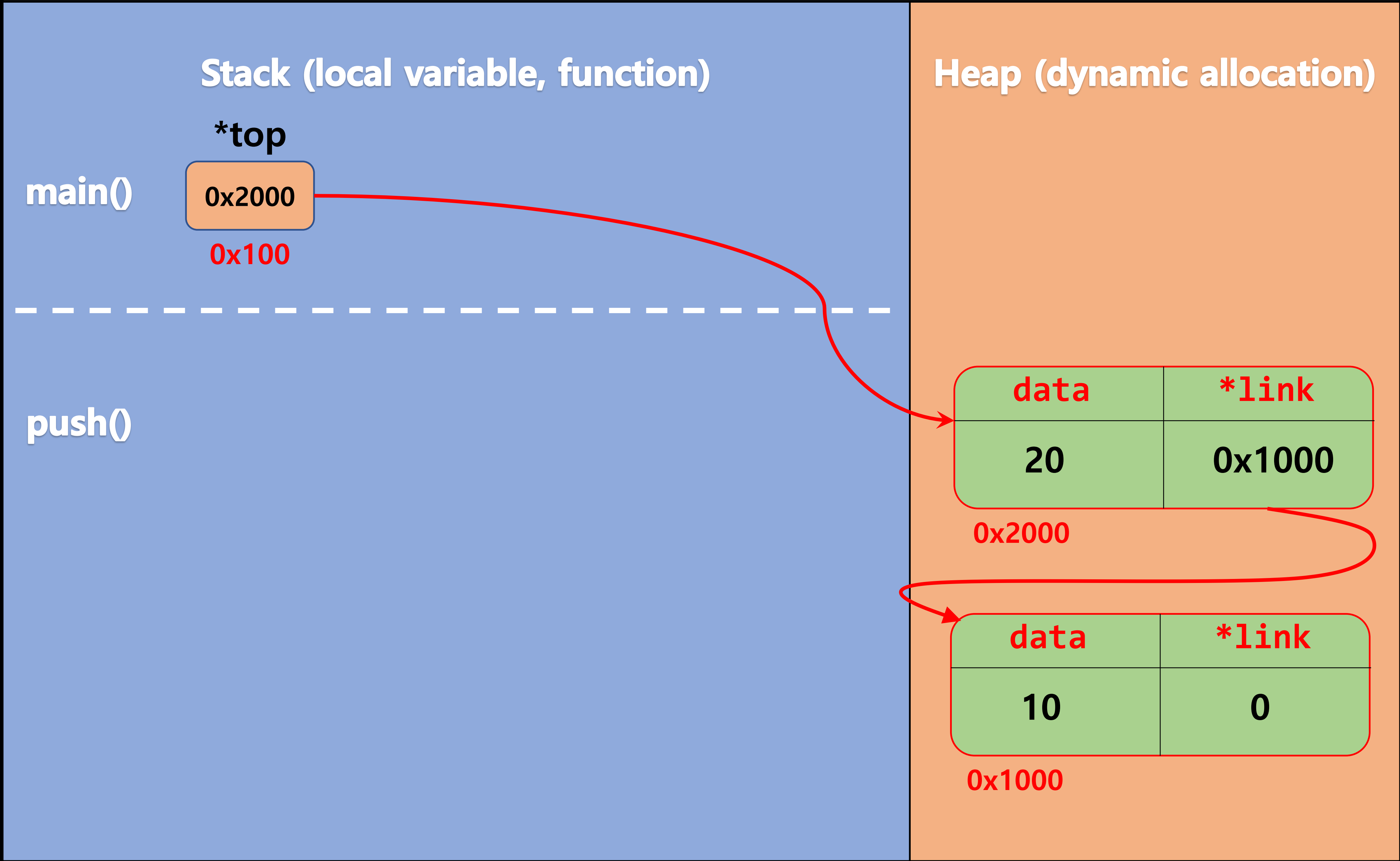
0x1000

Data Struct

Stack

code)

```
push(&top, 30);
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x2000

0x100

push()

data	*link
20	0x1000

0x2000

data	*link
10	0

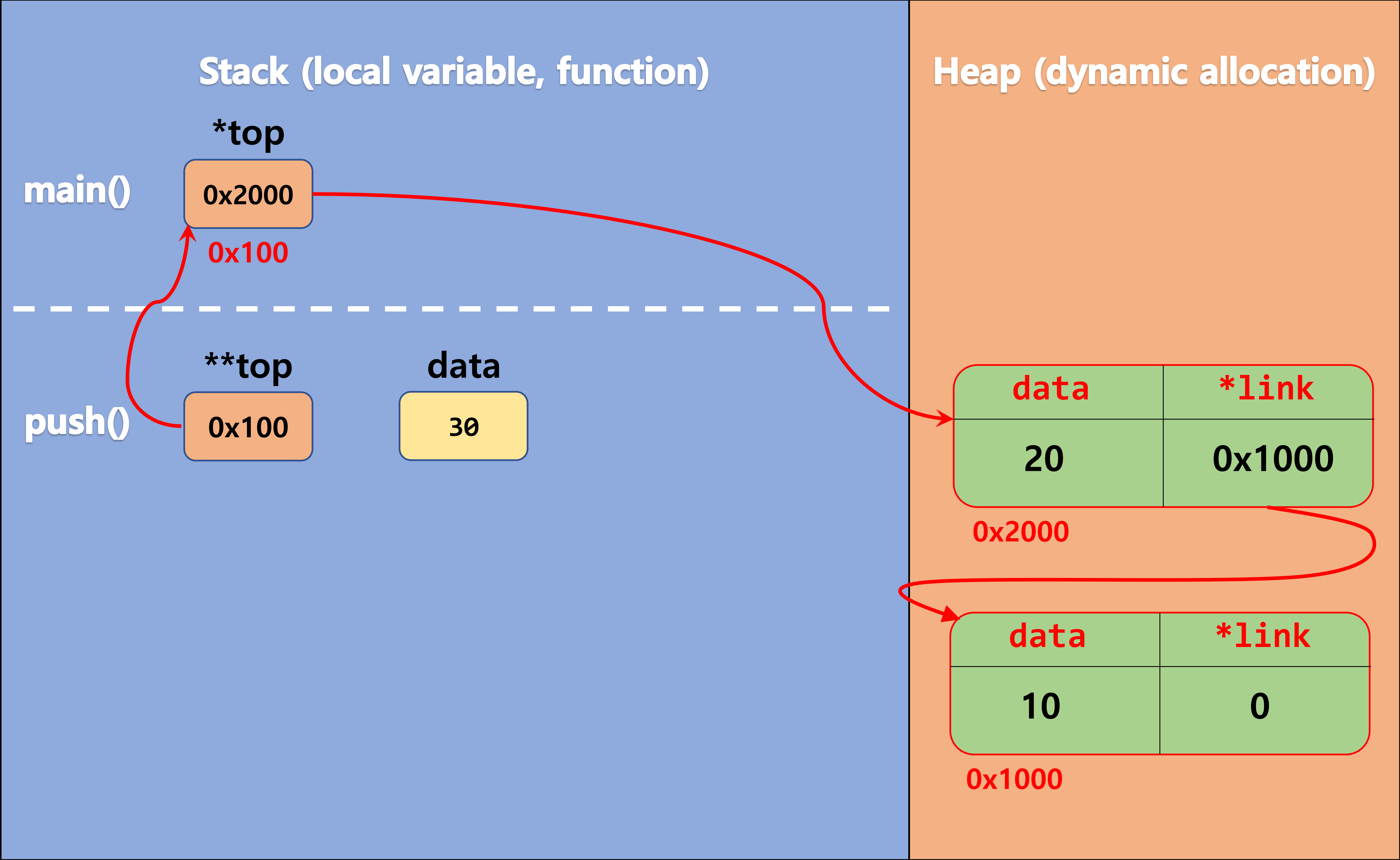
0x1000

Data Struct

Stack

code)

```
push(&top, 30);
```

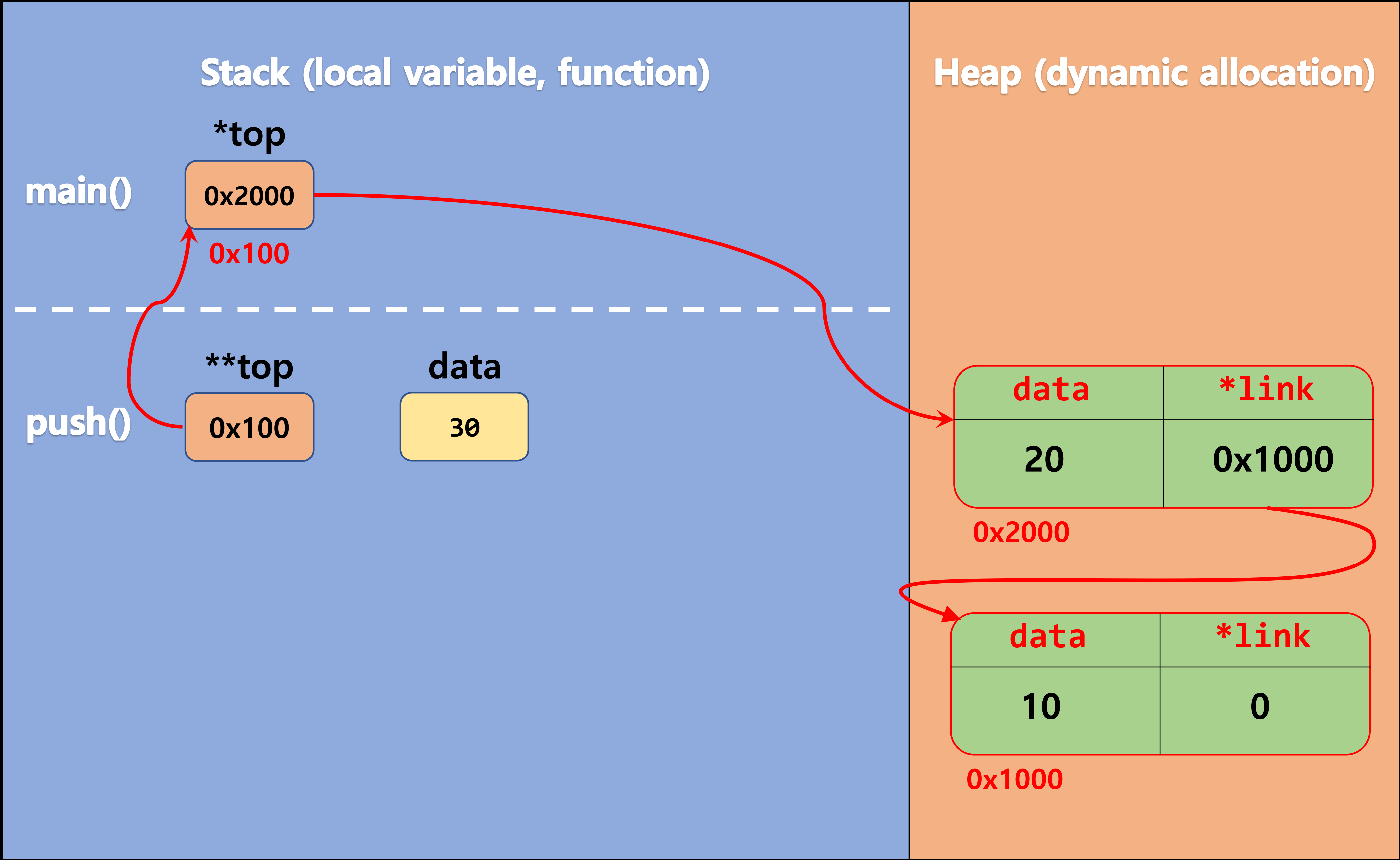


Data Struct

Stack

code)

```
Stack* tmp = *top;
```

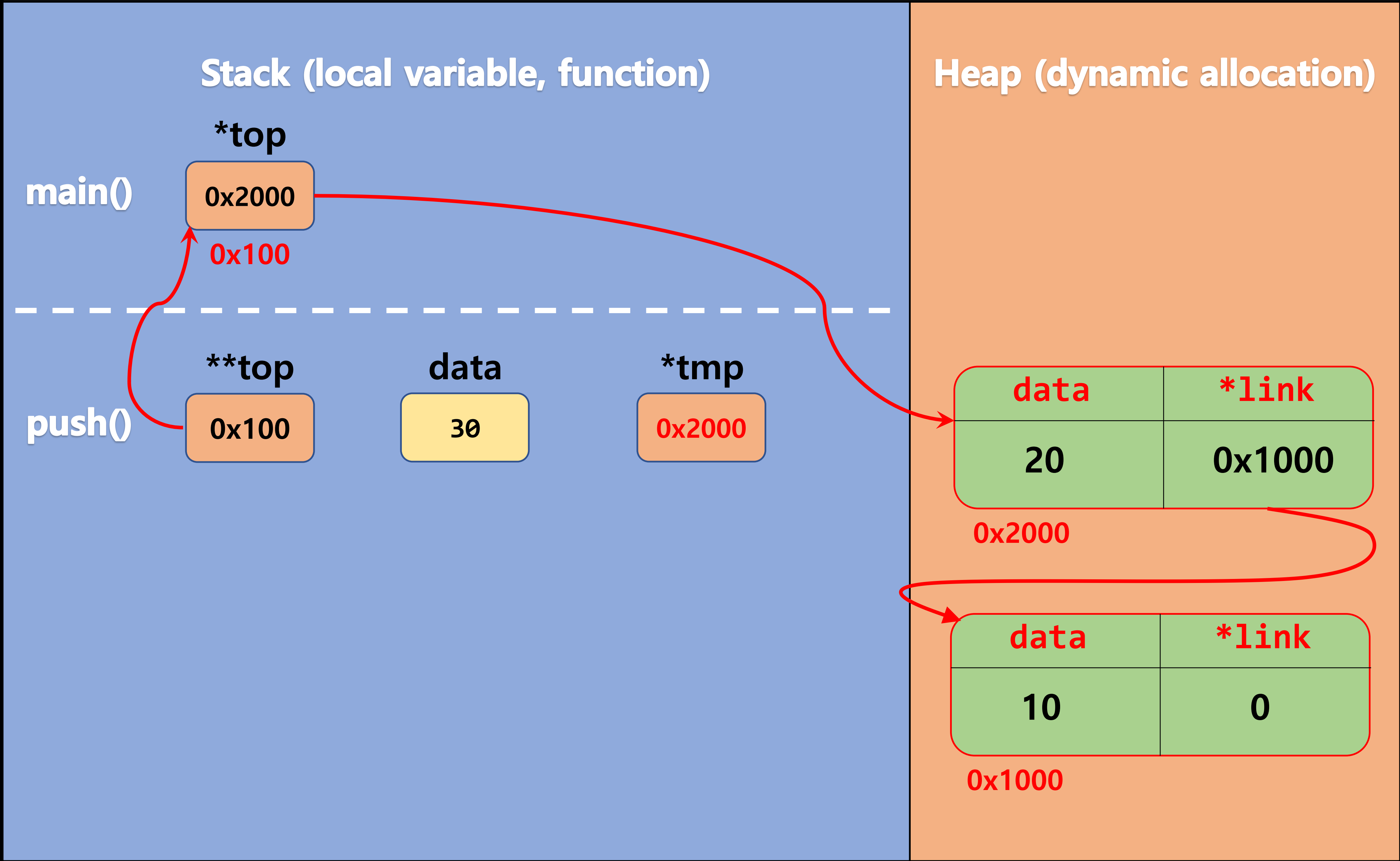


Data Struct

Stack

code)

```
Stack* tmp = *top;
```



Stack (local variable, function)

main()

*top

0x2000

0x100

push()

**top

0x100

data

30

*tmp

0x2000

Heap (dynamic allocation)

data	*link
20	0x1000

0x2000

data	*link
10	0

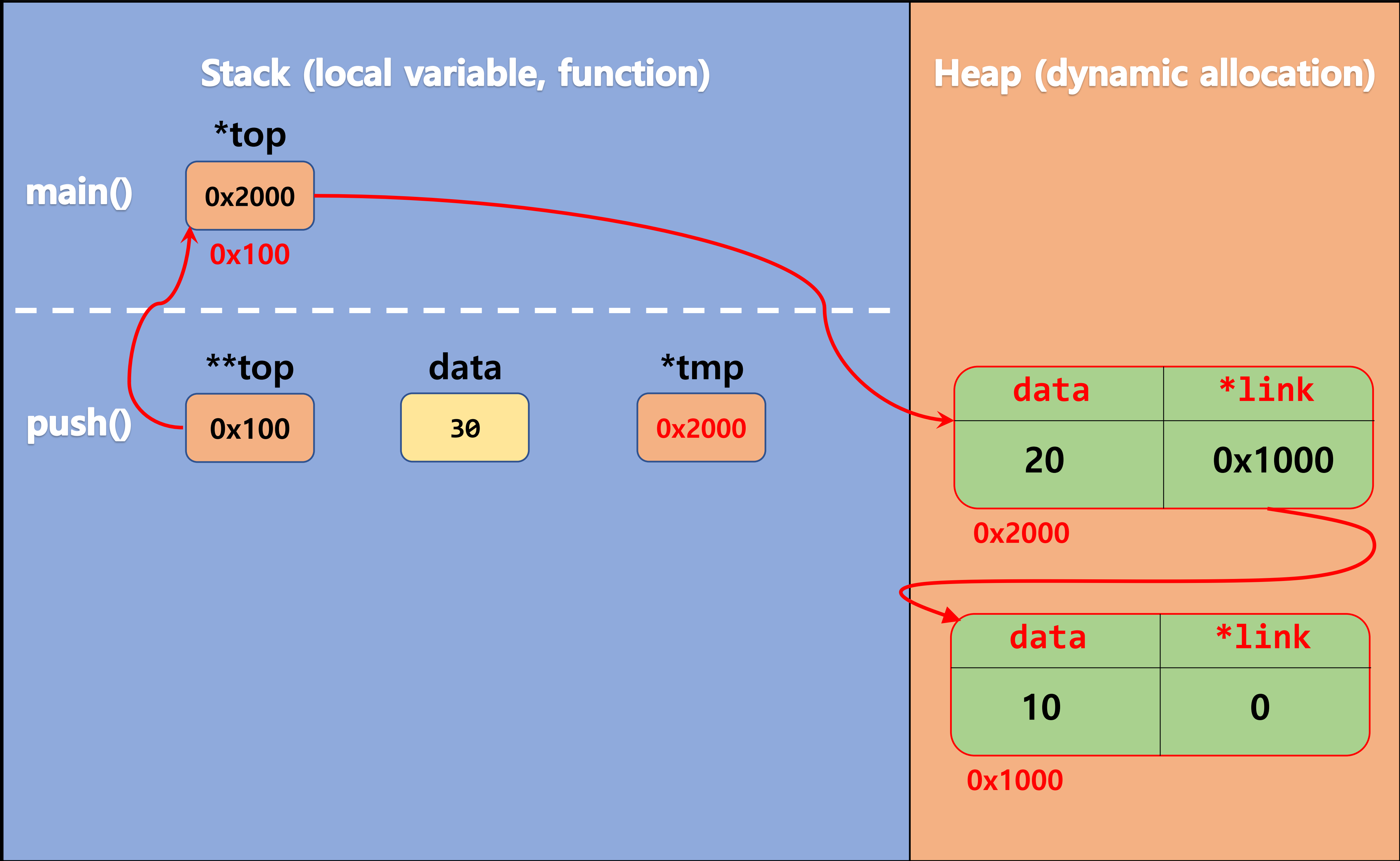
0x1000

Data Struct

Stack

code)

```
*top = getNode();
```

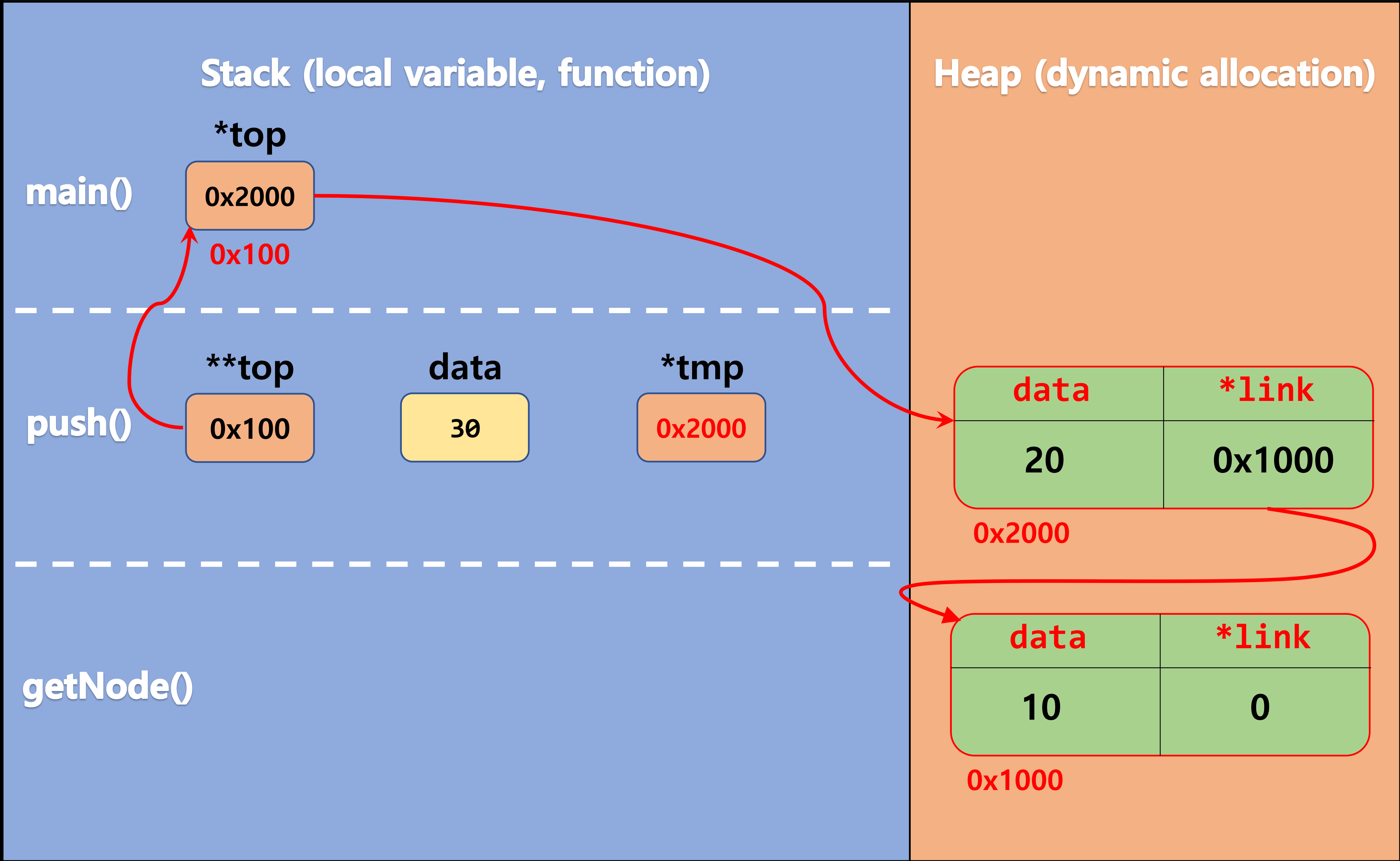


Data Struct

Stack

code)

```
*top = getNode();
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x2000

0x100

push()

**top

0x100

data

30

*tmp

0x2000

getNode()

data	*link
20	0x1000

0x2000

data	*link
10	0

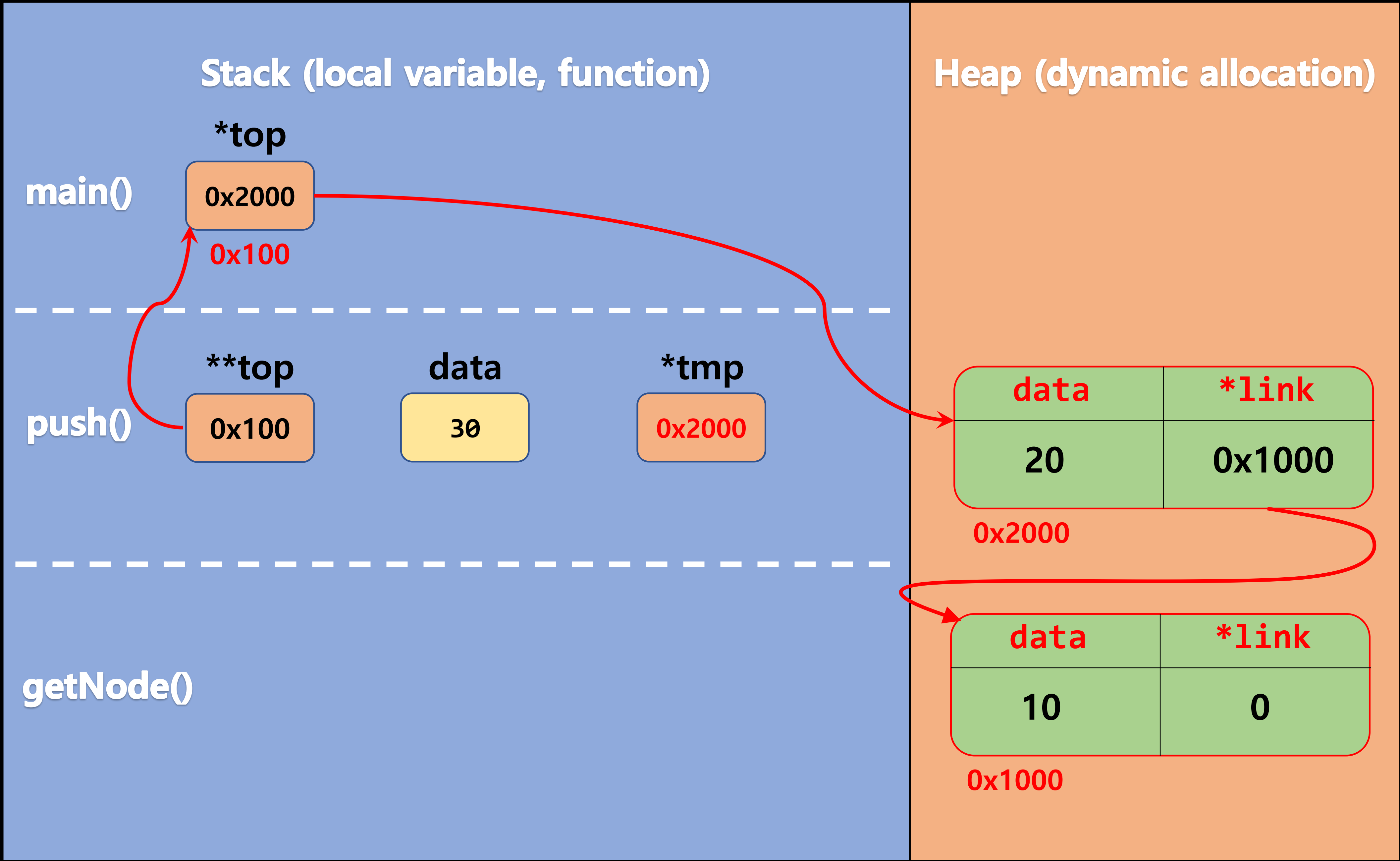
0x1000

Data Struct

Stack

code)

```
Stack* newNode = (Stack*)malloc(sizeof(Stack));
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x2000

0x100

push()

**top

0x100

data

30

*tmp

0x2000

getNode()

data	*link
20	0x1000

0x2000

data	*link
10	0

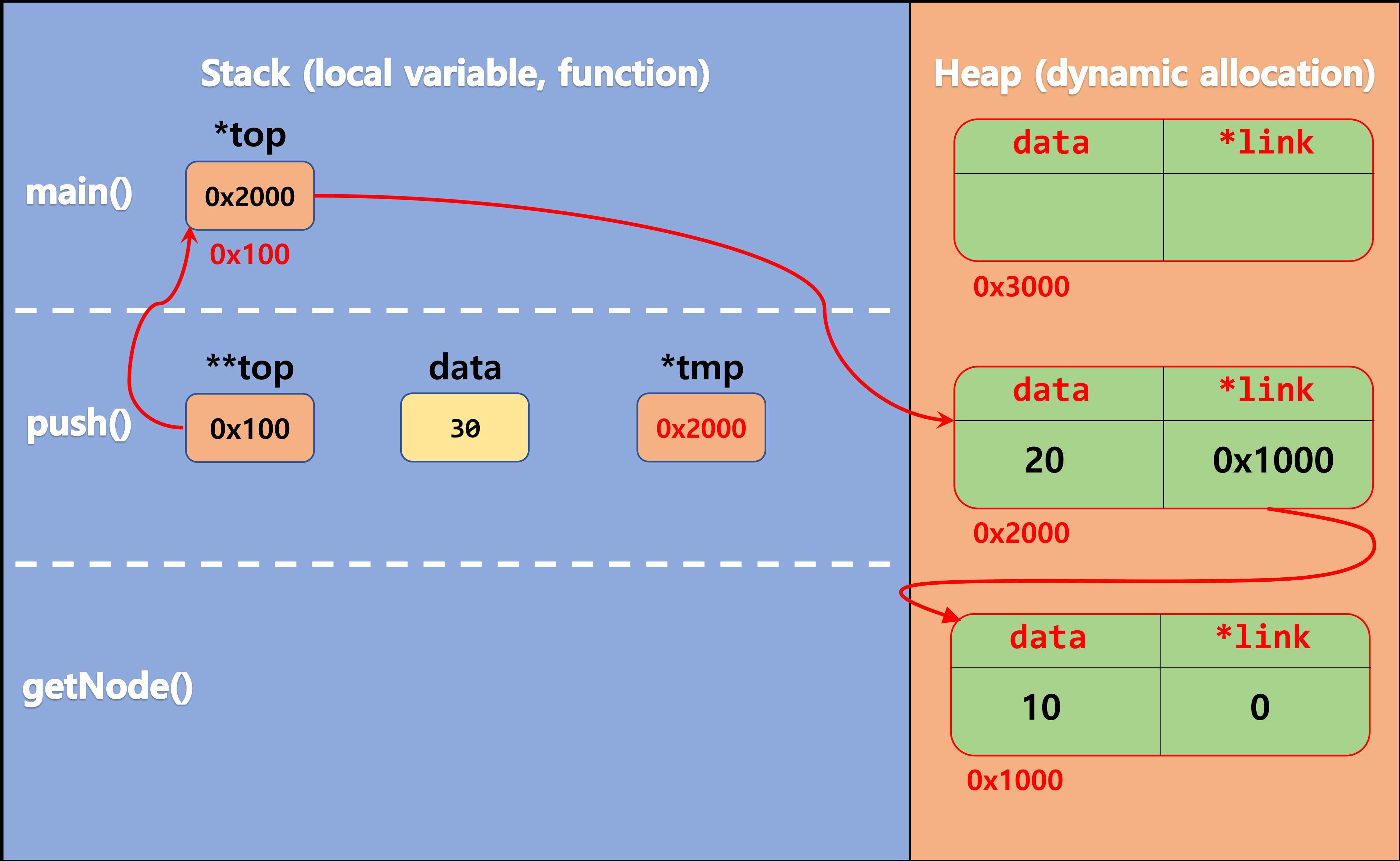
0x1000

Data Struct

Stack

code)

```
Stack* newNode = (Stack*)malloc(sizeof(Stack));
```

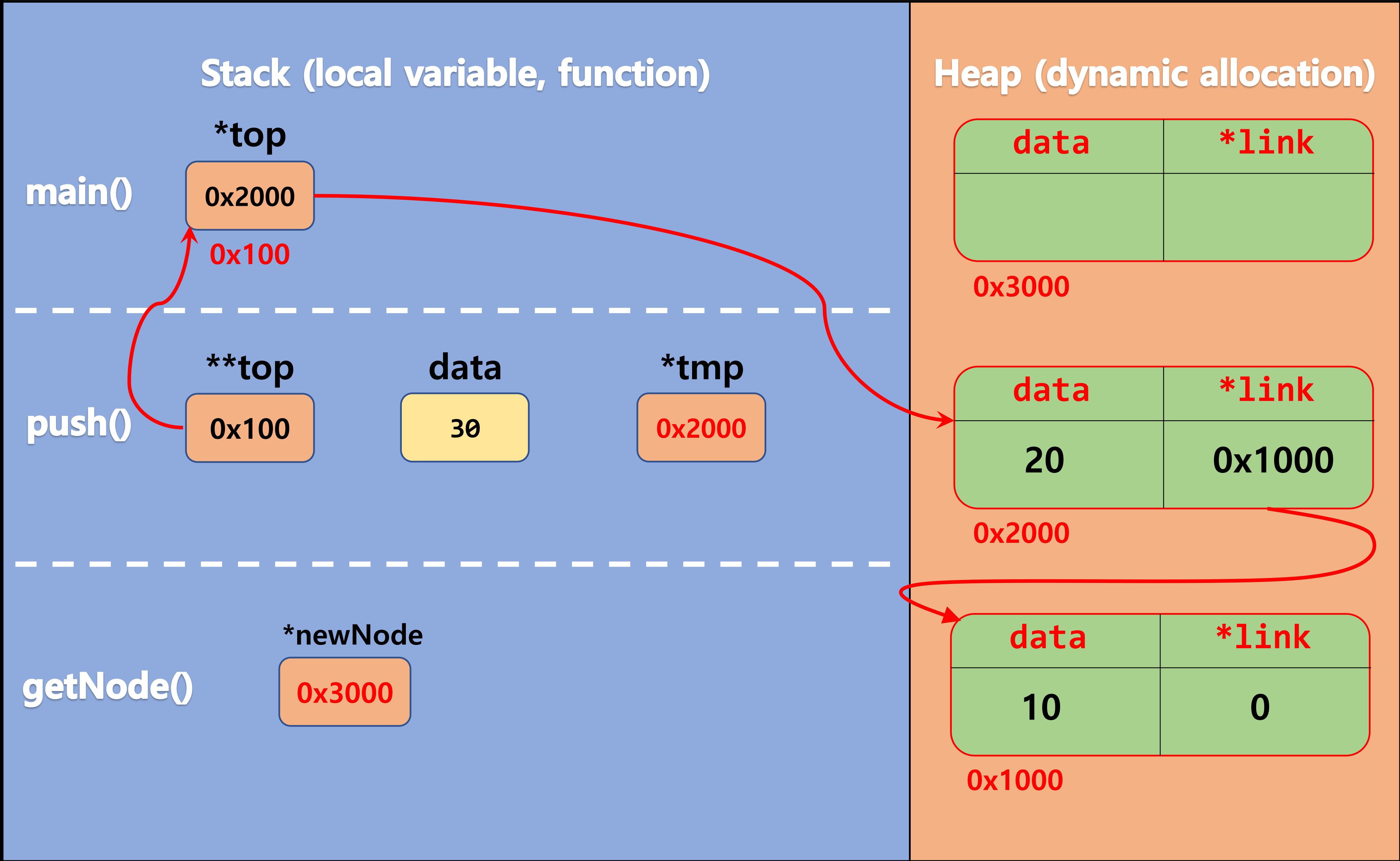


code)

```c
Stack* newNode = (Stack*)malloc(sizeof(Stack));
```

Data Struct

Stack

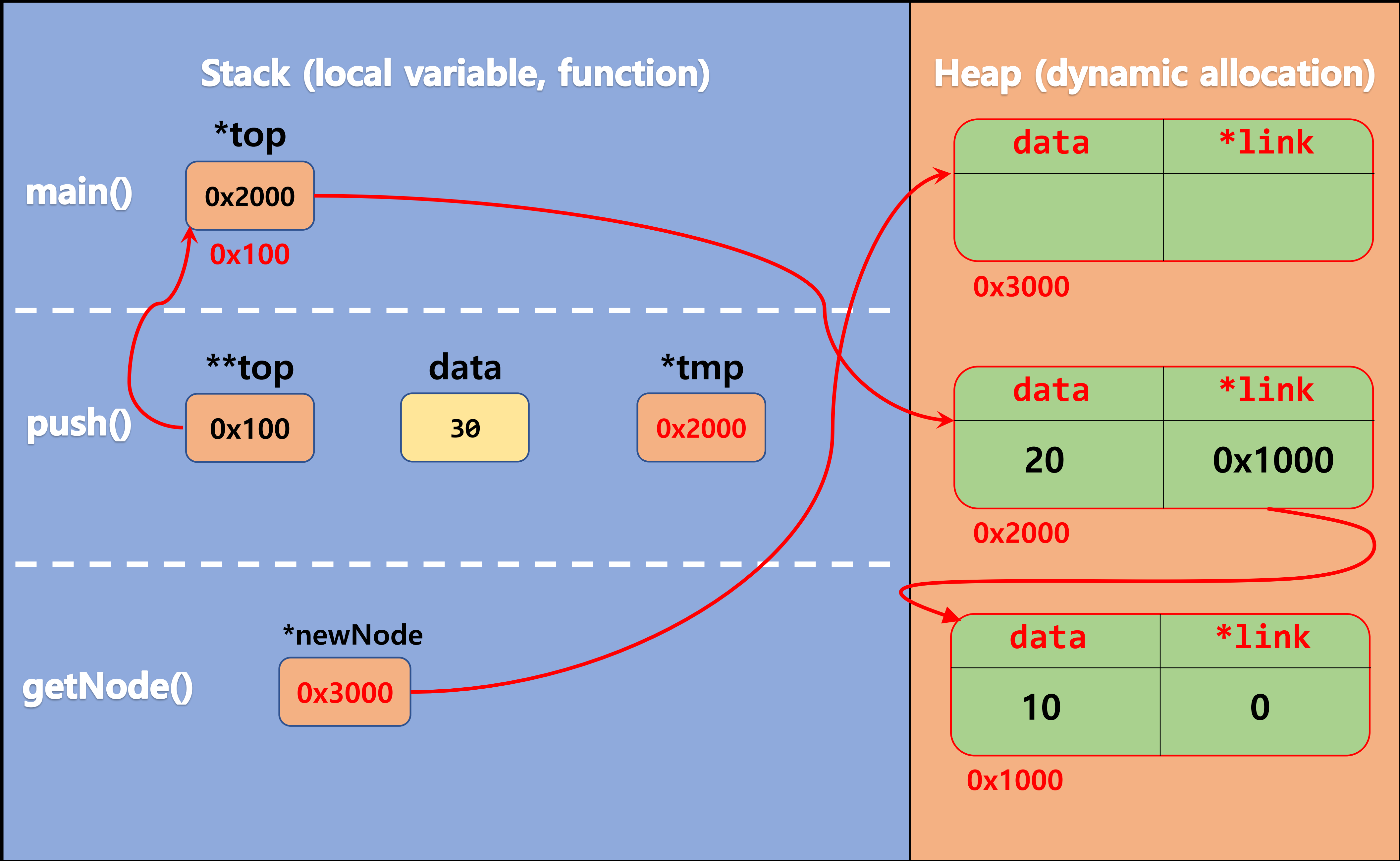


Data Struct

Stack

code)

```
Stack* newNode = (Stack*)malloc(sizeof(Stack));
```

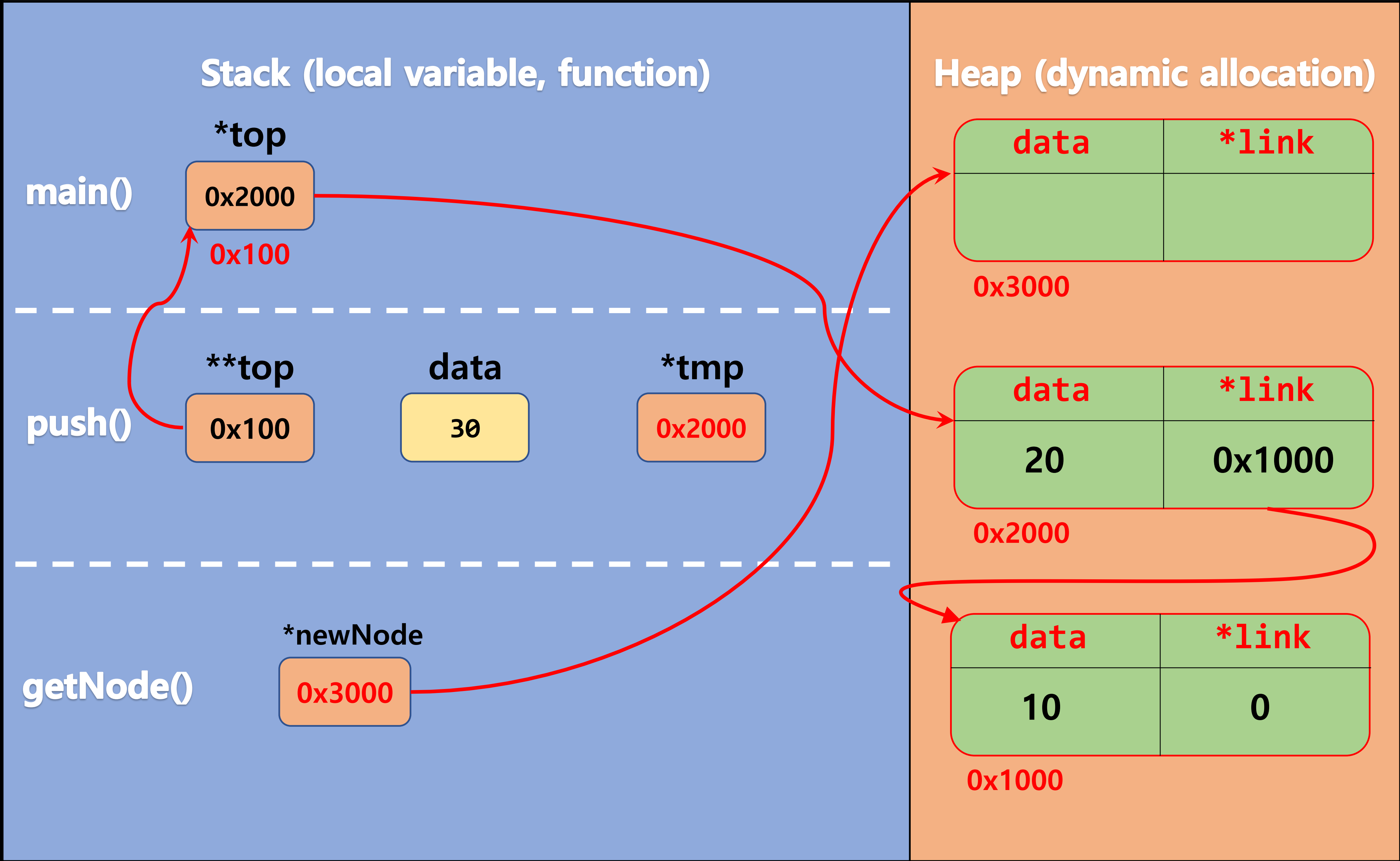


Data Struct

Stack

code)

```
newNode->link = NULL;
```

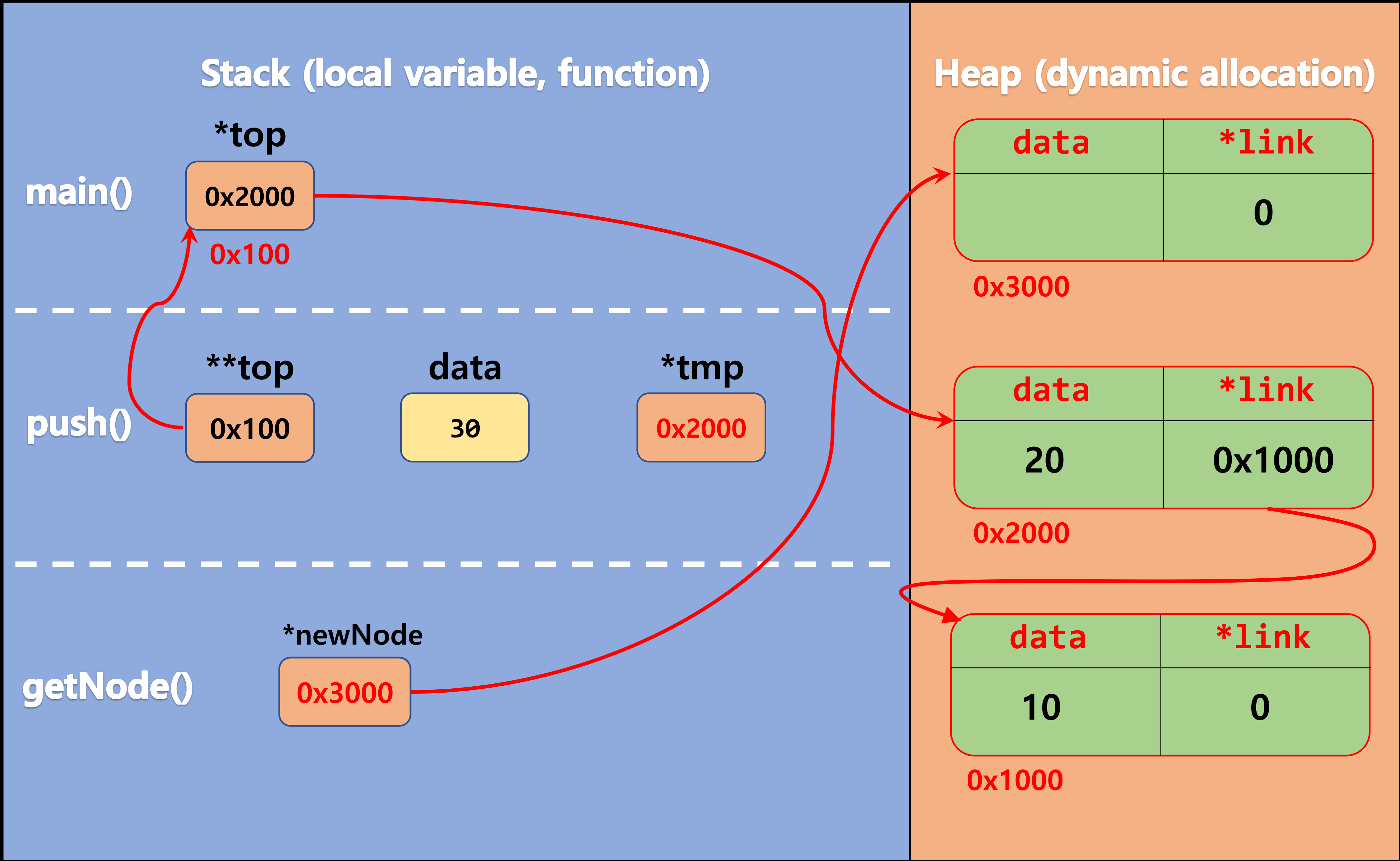


Data Struct

Stack

code)

```
newNode->link = NULL;
```

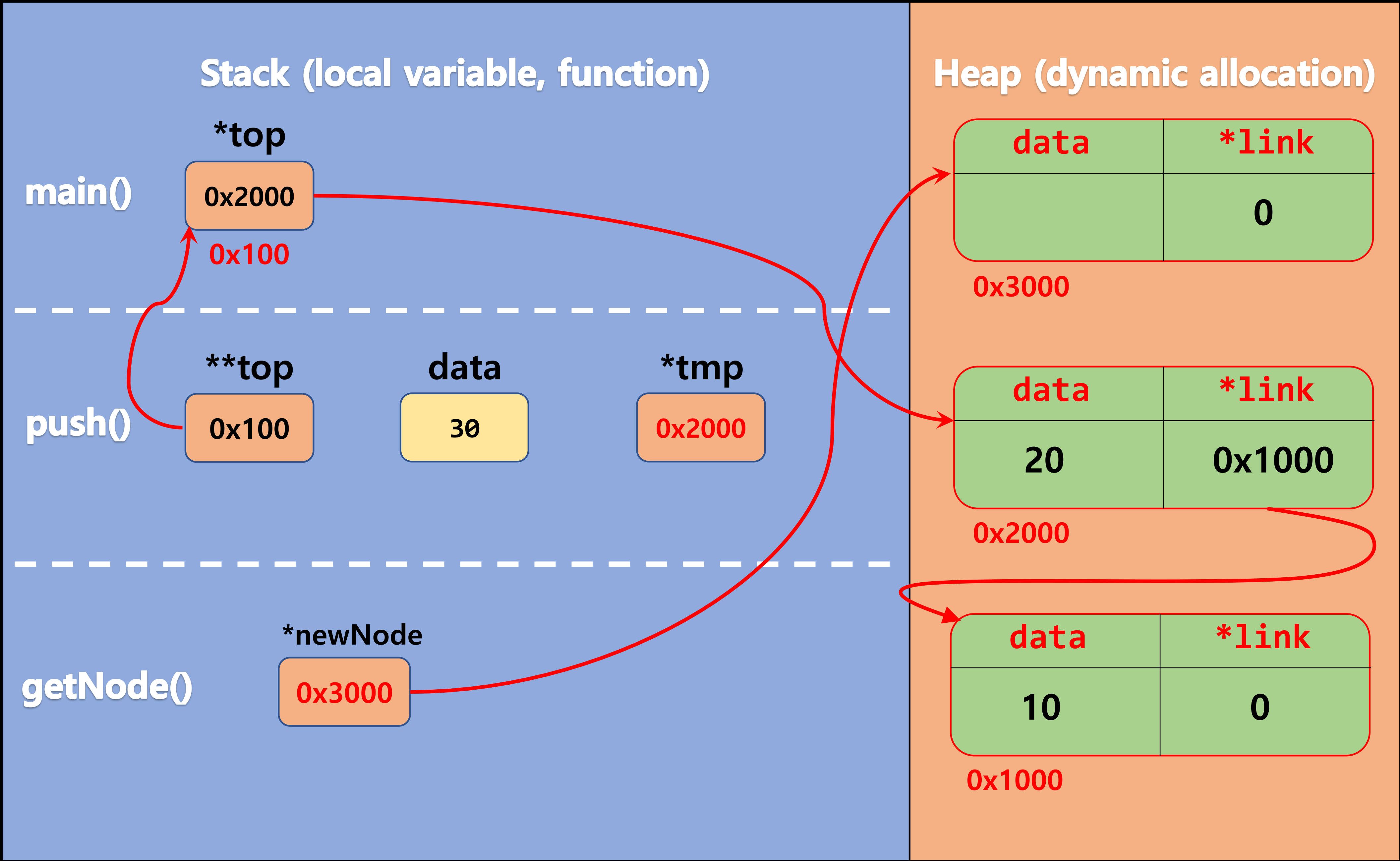


Data Struct

Stack

code)

```
return newNode;
```

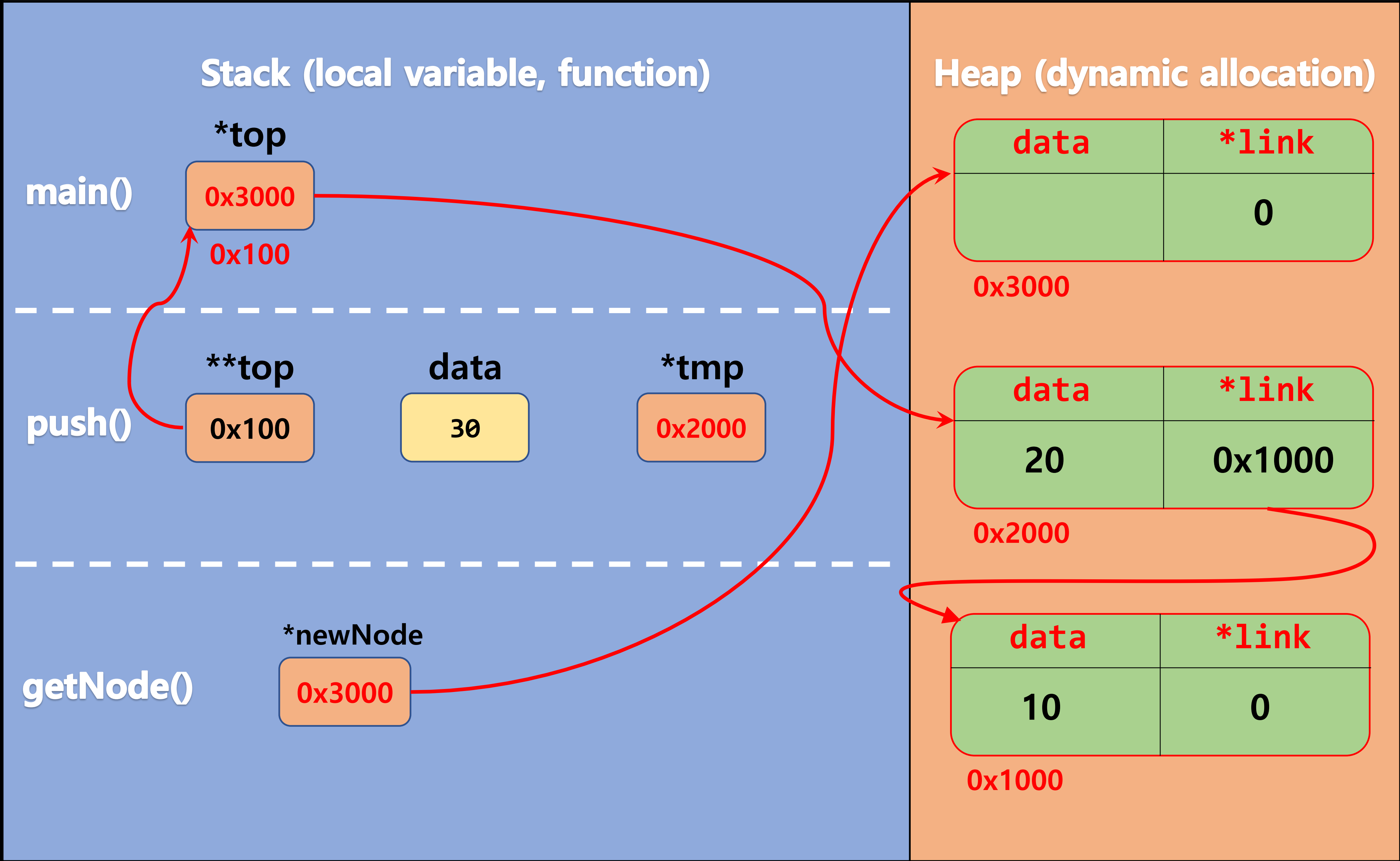


Data Struct

Stack

code)

```
return newNode;
```



Stack (local variable, function)

main()

*top

0x3000

0x100

push()

**top

0x100

data

30

*tmp

0x2000

getNode()

*newNode

0x3000

Heap (dynamic allocation)

data	*link
	0

0x3000

data	*link
20	0x1000

0x2000

data	*link
10	0

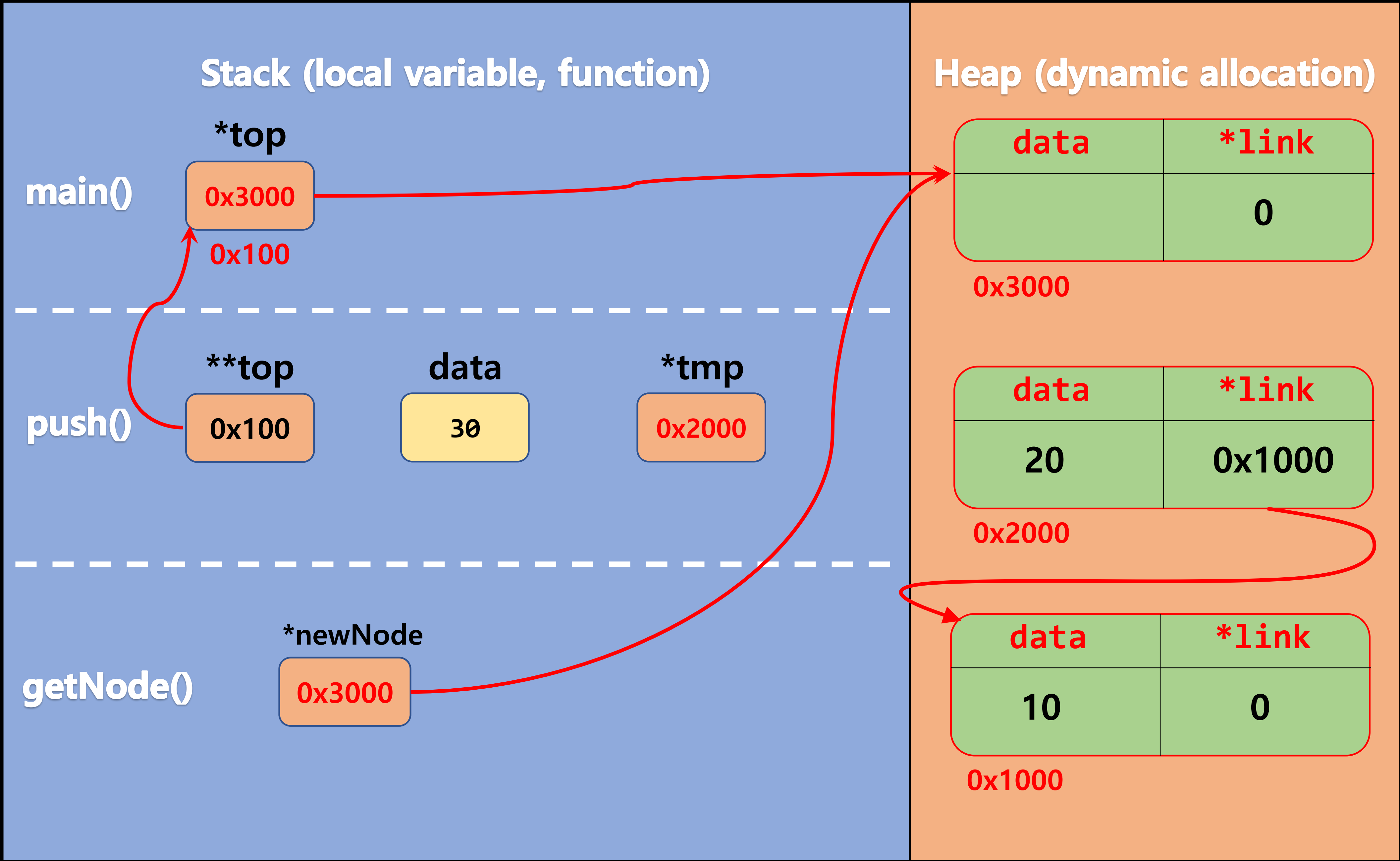
0x1000

Data Struct

Stack

code)

```
return newNode;
```



Stack (local variable, function)

main()

*top

0x3000

0x100

push()

**top

0x100

data

30

*tmp

0x2000

getNode()

*newNode

0x3000

Heap (dynamic allocation)

data	*link
	0

0x3000

data	*link
20	0x1000

0x2000

data	*link
10	0

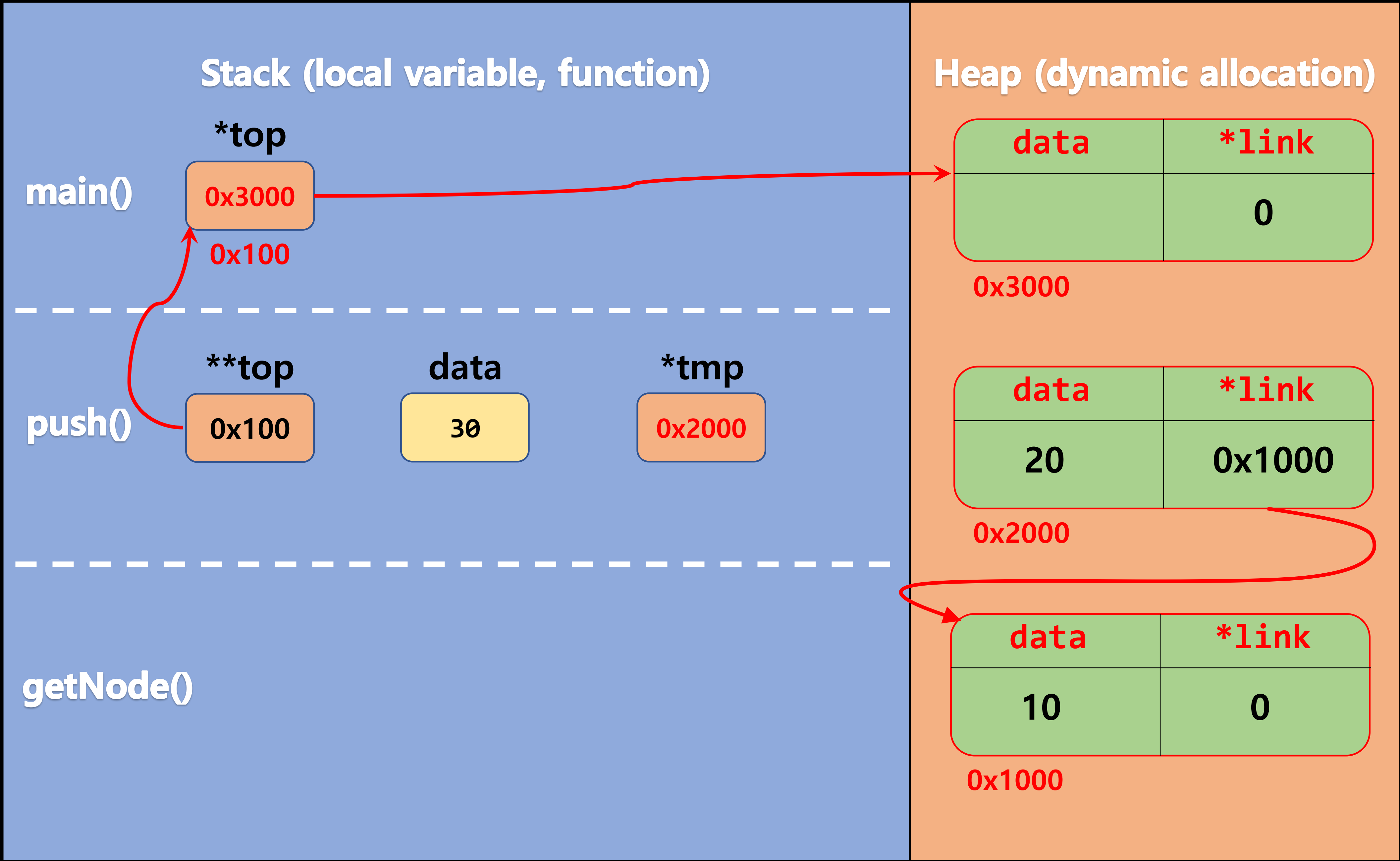
0x1000

Data Struct

Stack

code)

```
return newNode;
```



Stack (local variable, function)

main()

*top

0x3000

0x100

push()

**top

0x100

data

30

*tmp

0x2000

getNode()

Heap (dynamic allocation)

data	*link
	0

0x3000

data	*link
20	0x1000

0x2000

data	*link
10	0

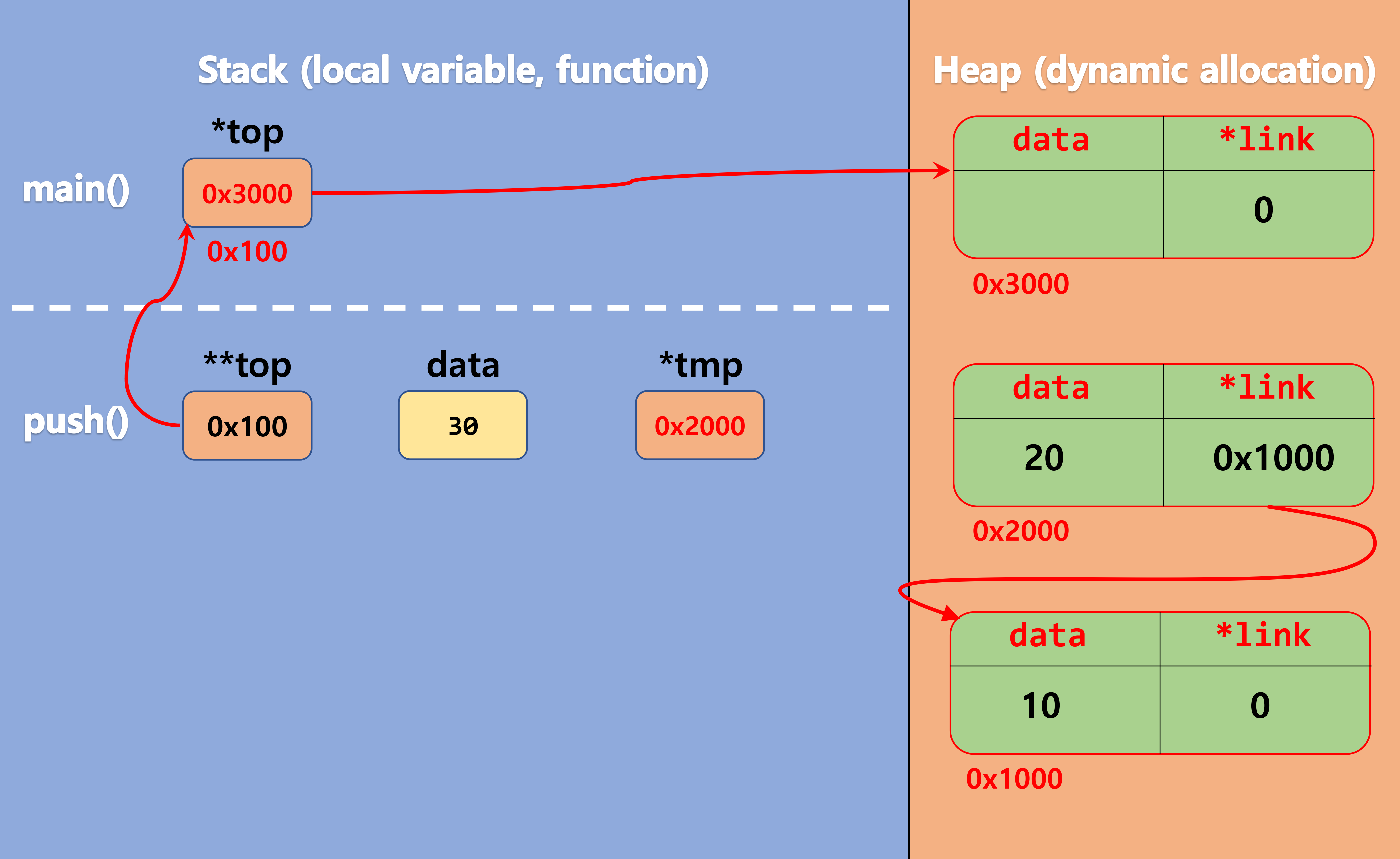
0x1000

Data Struct

Stack

code)

```
return newNode;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x3000

0x100

push()

**top

0x100

data

30

*tmp

0x2000

data	*link
	0

0x3000

data	*link
20	0x1000

0x2000

data	*link
10	0

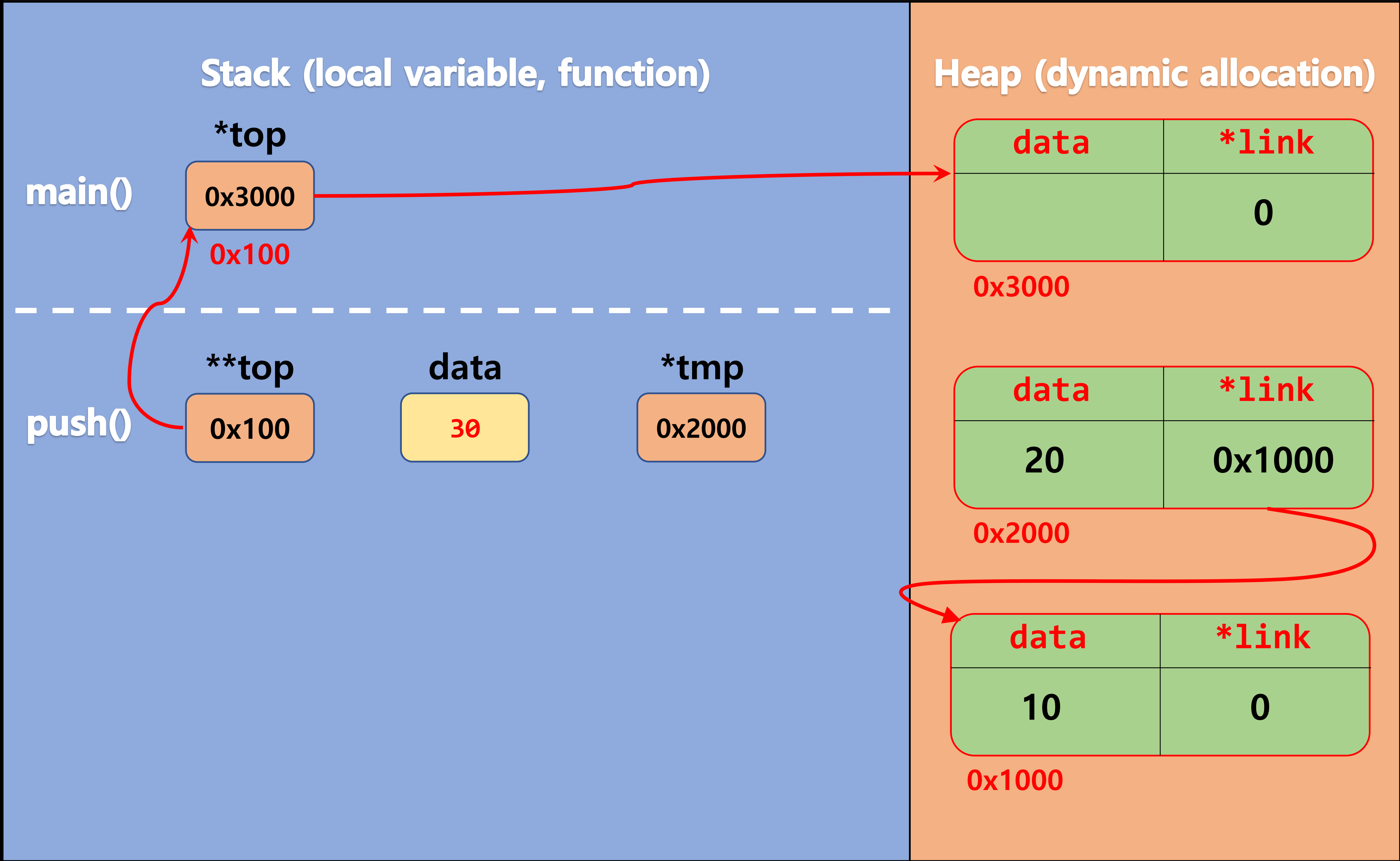
0x1000

Data Struct

Stack

code)

```
(*top)->data = data;
```



Stack (local variable, function)

main()

*top

0x3000

0x100

push()

**top

0x100

data

30

*tmp

0x2000

Heap (dynamic allocation)

data	*link
	0

0x3000

data	*link
20	0x1000

0x2000

data	*link
10	0

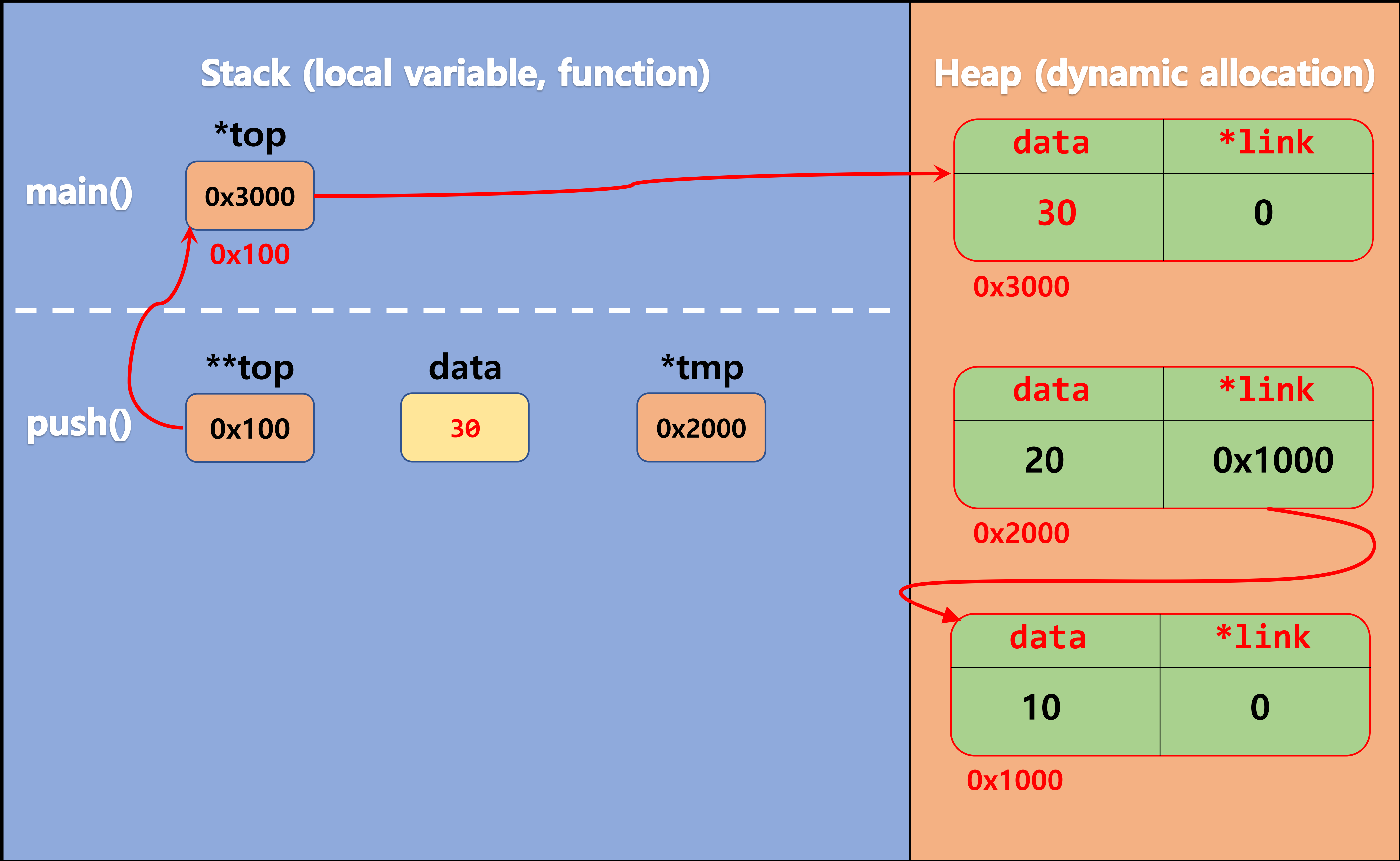
0x1000

Data Struct

Stack

code)

```
(*top)->data = data;
```



Stack (local variable, function)

Heap (dynamic allocation)

main()

*top

0x3000

0x100

push()

**top

0x100

data

30

*tmp

0x2000

data	*link
30	0

0x3000

data	*link
20	0x1000

0x2000

data	*link
10	0

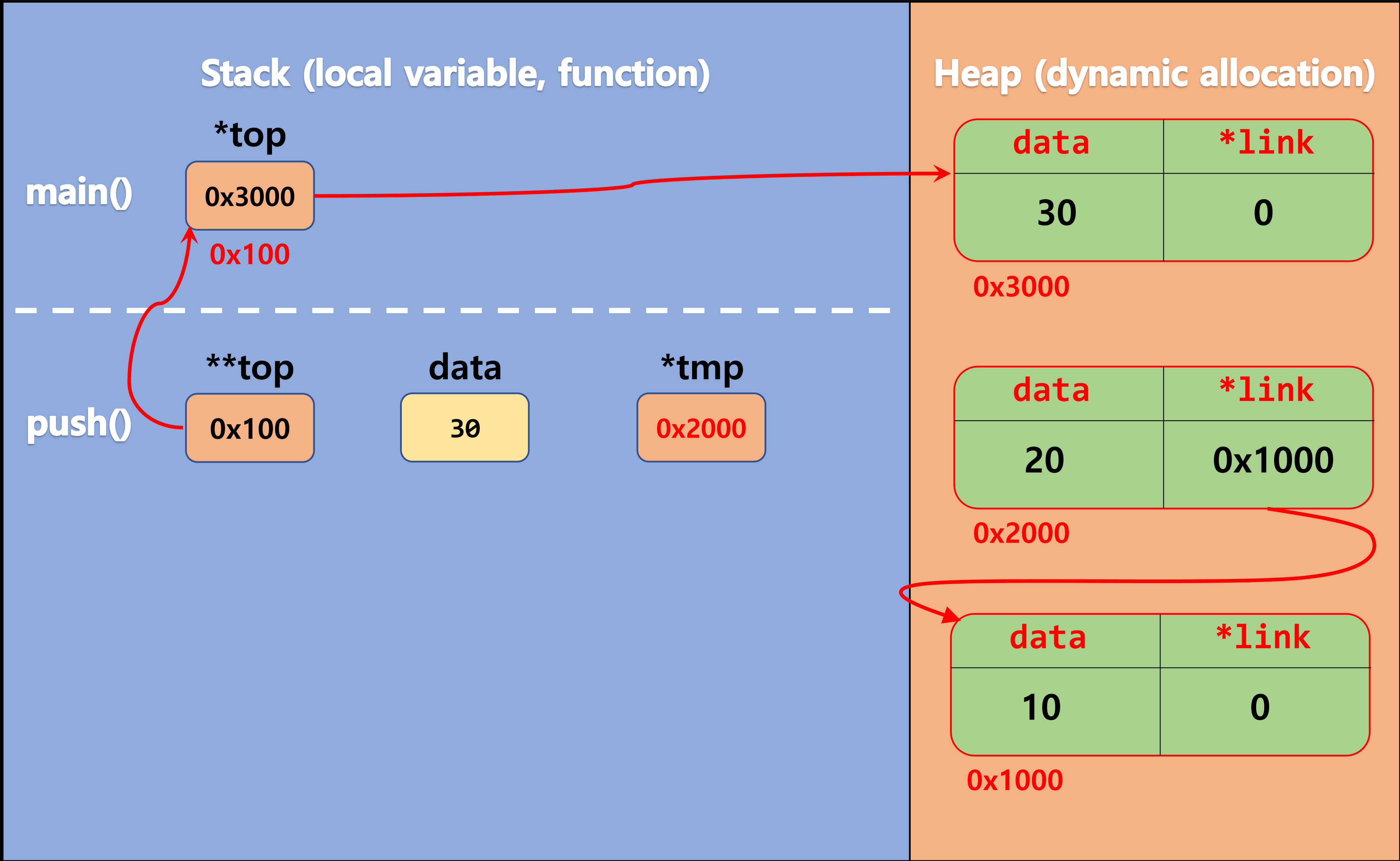
0x1000

Data Struct

Stack

code)

```
(*top)->link = tmp;
```

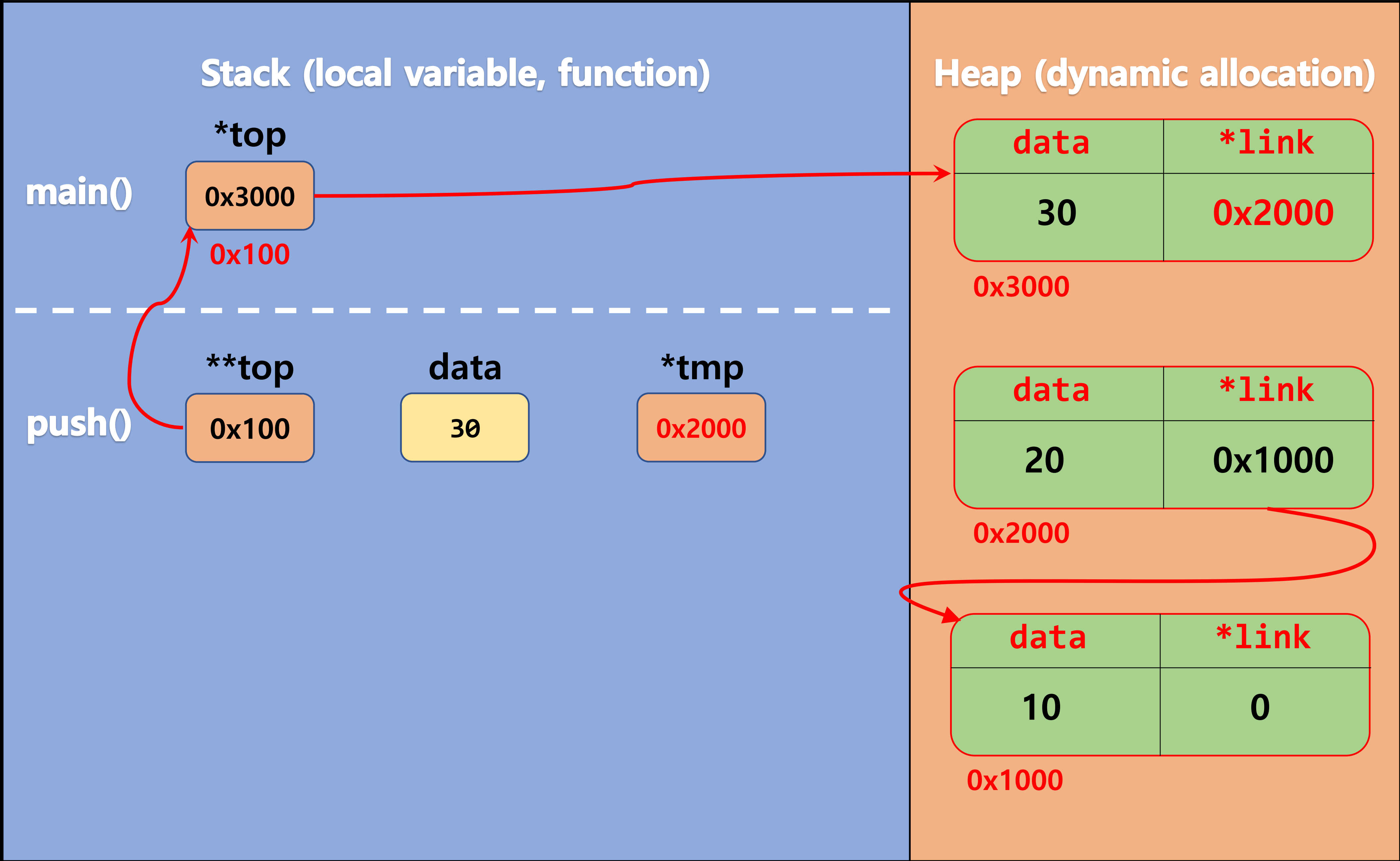


Data Struct

Stack

code)

```
(*top)->link = tmp;
```

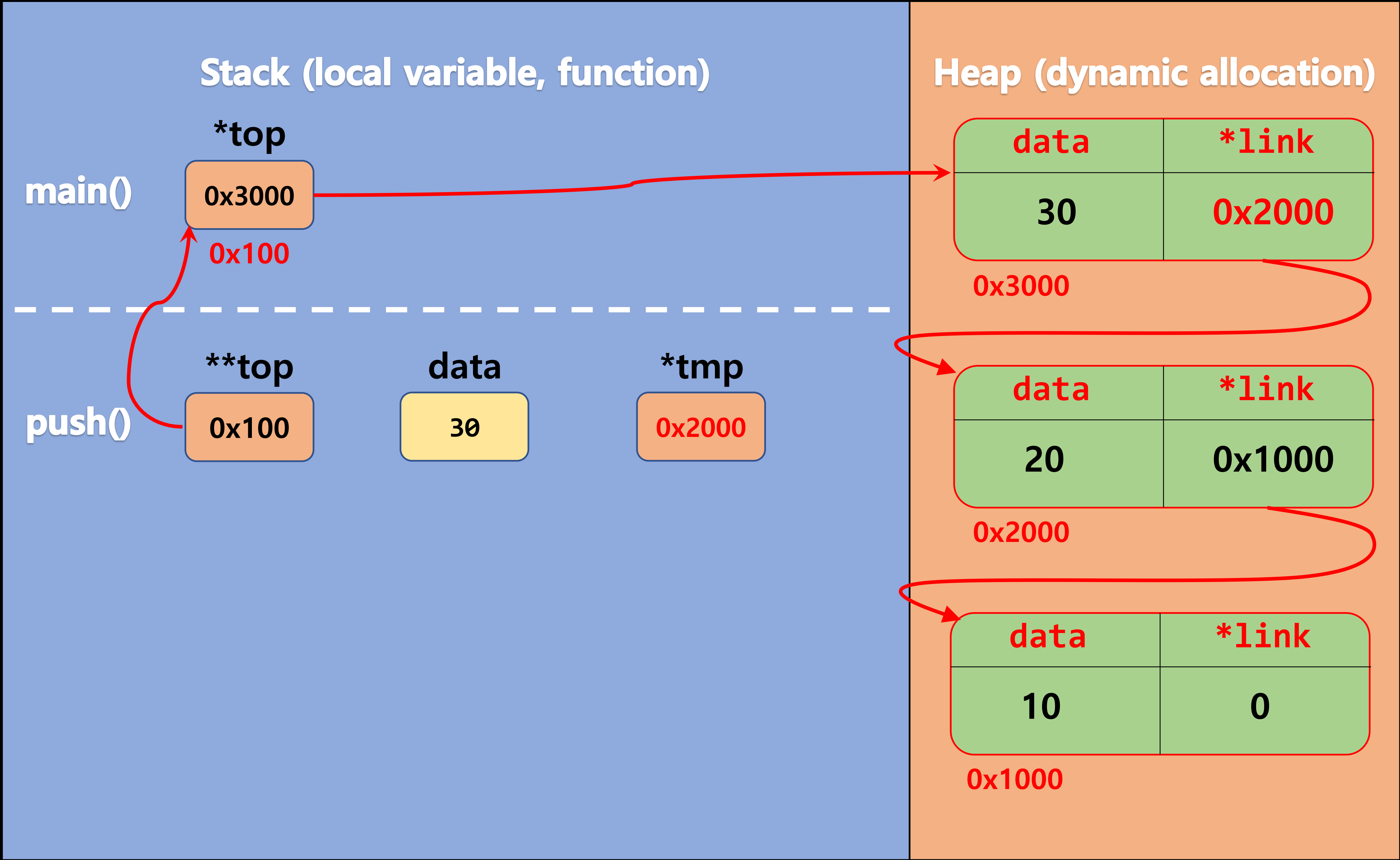


Data Struct

Stack

code)

```
(*top)->link = tmp;
```



Stack (local variable, function)

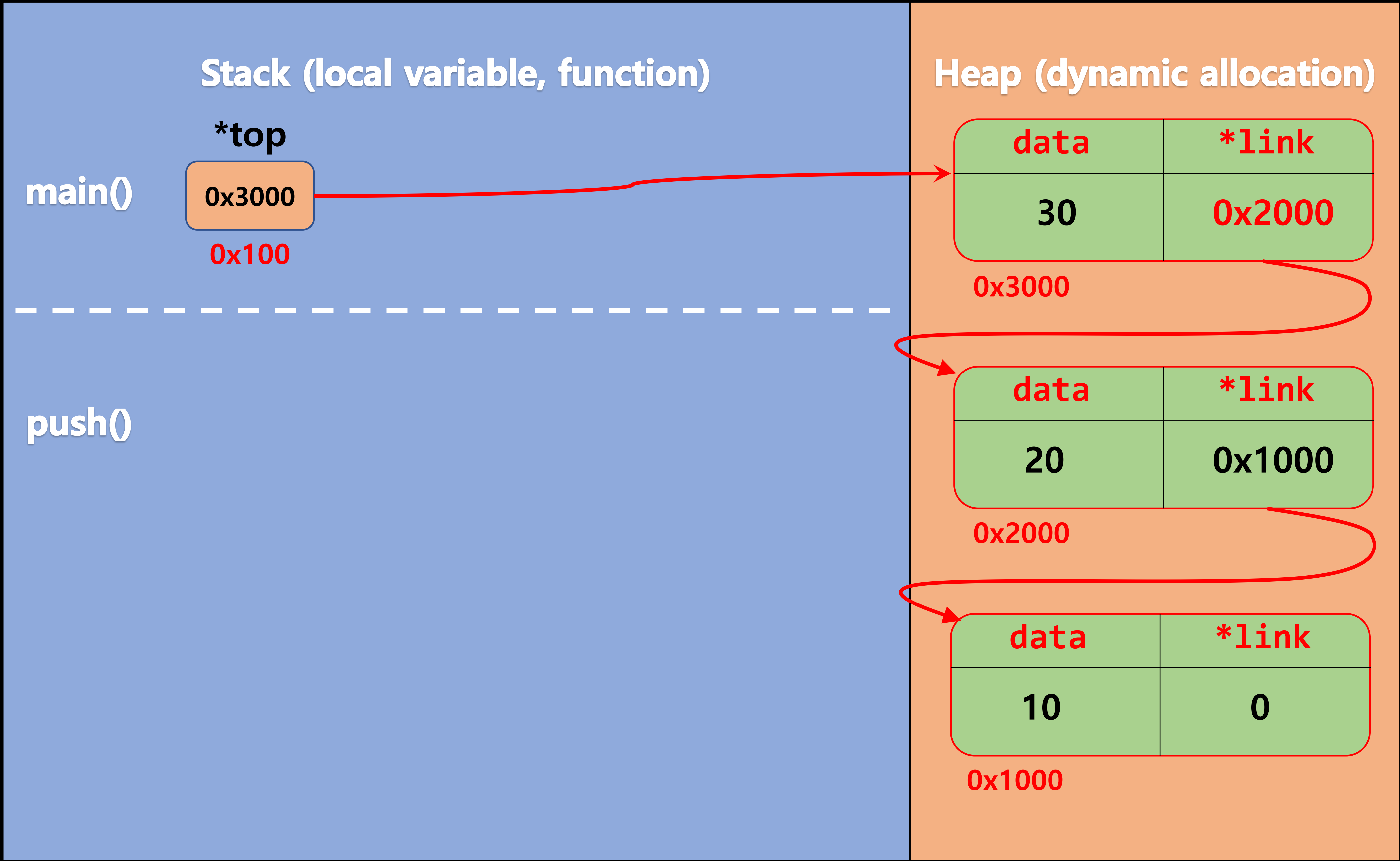
Heap (dynamic allocation)

Data Struct

Stack

code)

```
(*top)->link = tmp;
```



Stack (local variable, function)

main()

*top

0x3000

0x100

push()

Heap (dynamic allocation)

data	*link
30	0x2000

0x3000

data	*link
20	0x1000

0x2000

data	*link
10	0

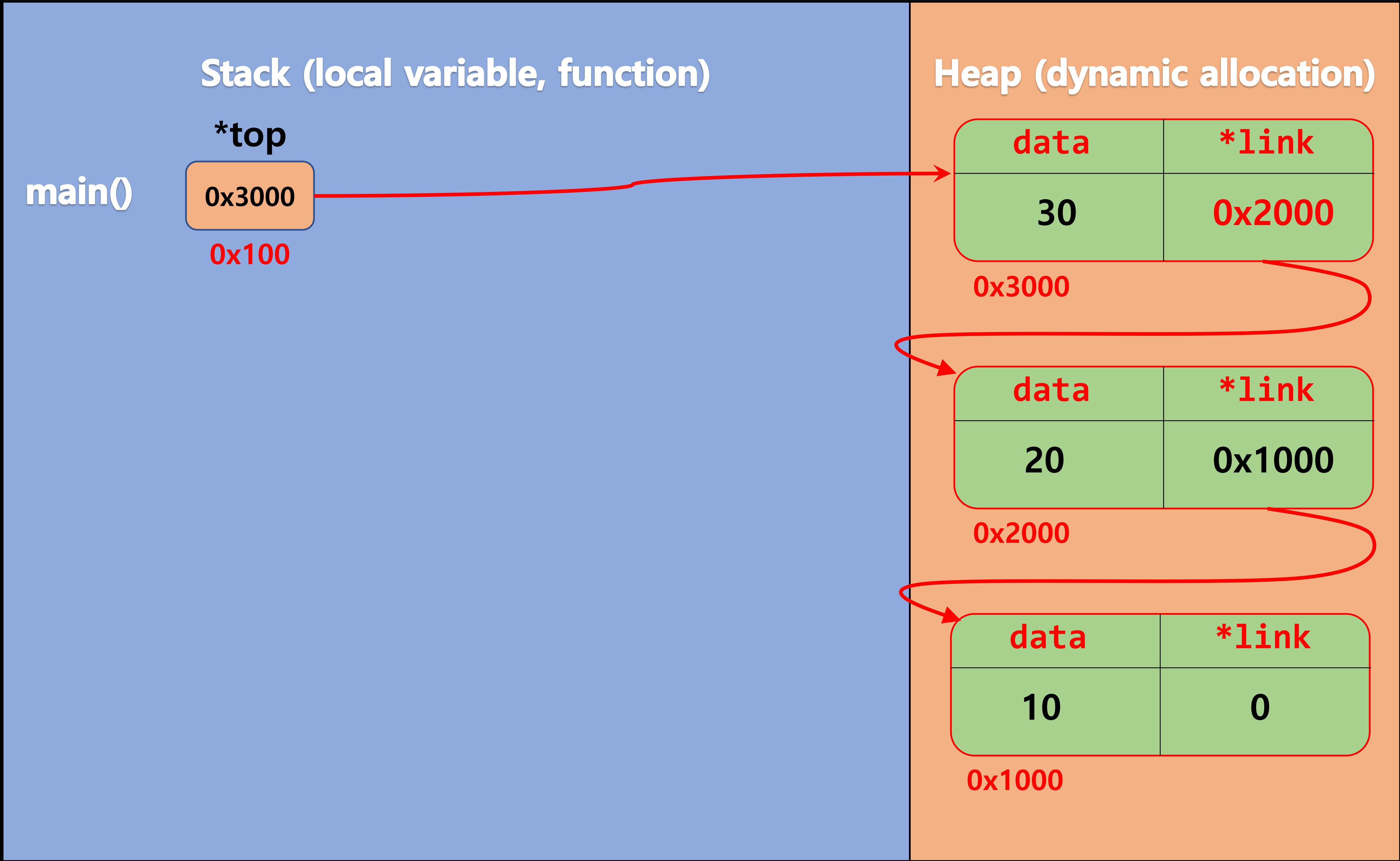
0x1000

Data Struct

Stack

code)

```
(*top)->link = tmp;
```



Stack (local variable, function)

main()

*top

0x3000

0x100

Heap (dynamic allocation)

data	*link
30	0x2000

0x3000

data	*link
20	0x1000

0x2000

data	*link
10	0

0x1000