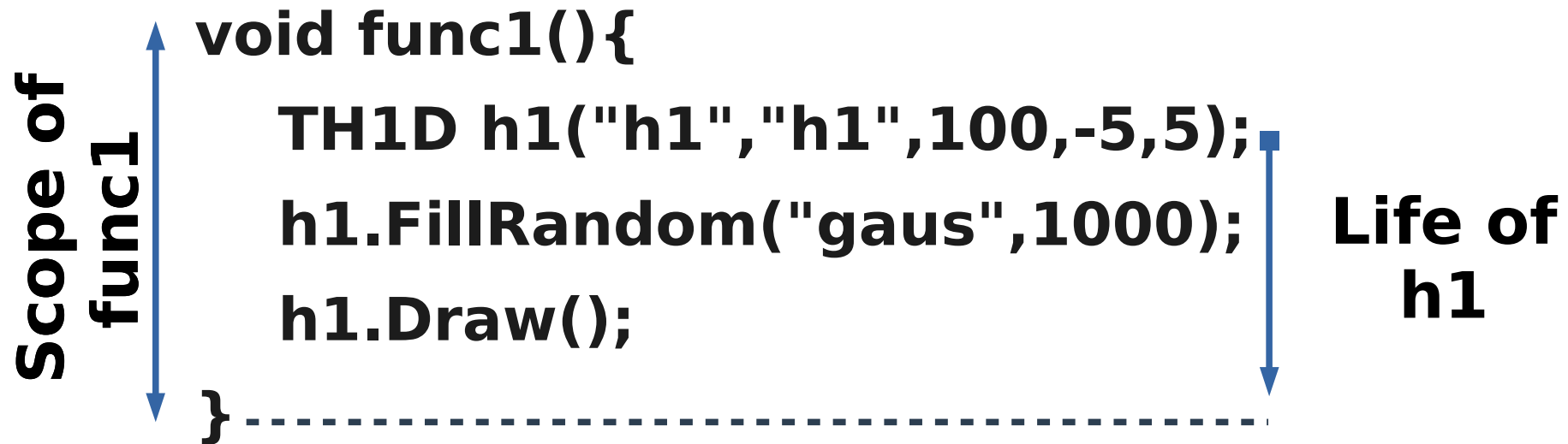


Variable Of Local Scope



h1 existed when it was Drawn

**h1 deleted from memory at end of func1
(disappears before we can see it)**

Utilising The Heap

Scope of func1

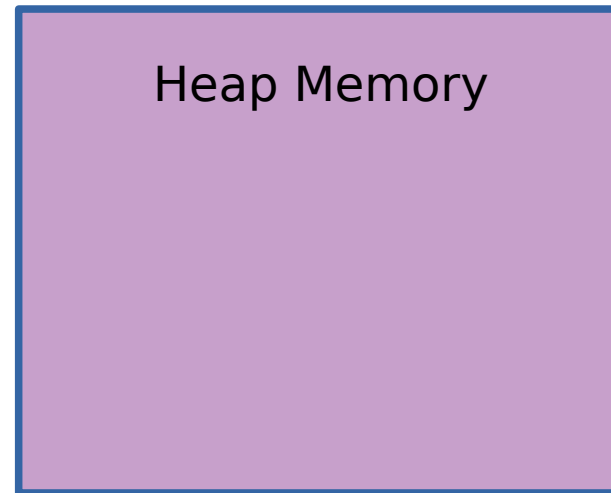
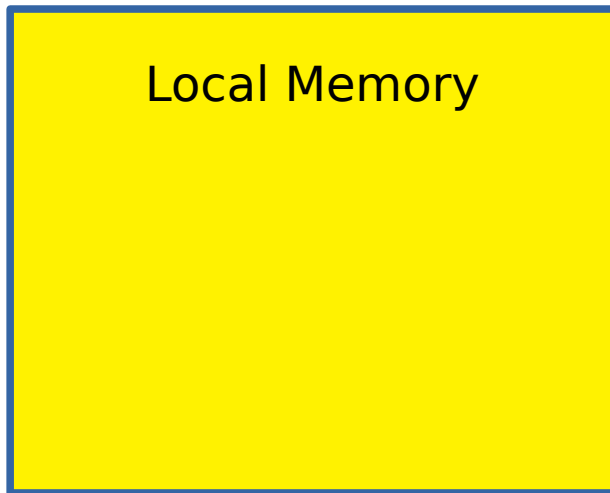
```
void func1(){  
    TH1D *hp = new TH1D("h1","h1",100,-5,5);  
    hp->FillRandom("gaus",1000);  
    hp->Draw();  
}
```

TH1D object - on heap
in memory until deleted

TH1D* pointer - hp - local
deleted at func1 end

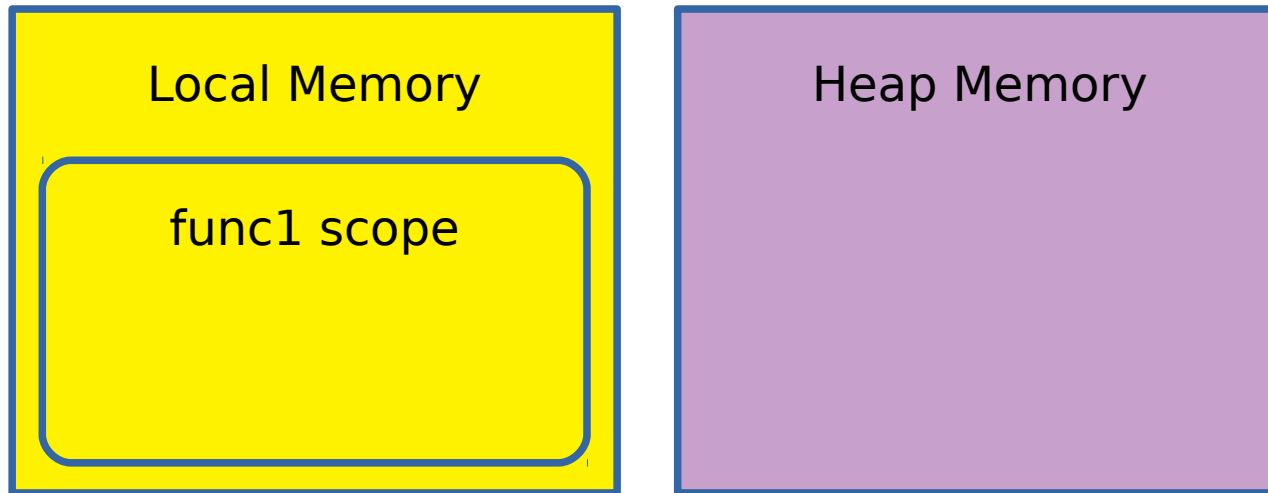
Utilising The Heap

```
void func1(){  
    TH1D *hp = new TH1D("h1","h1",100,-5,5);  
    hp->FillRandom("gaus",1000);  
    hp->Draw();  
}
```



Utilising The Heap

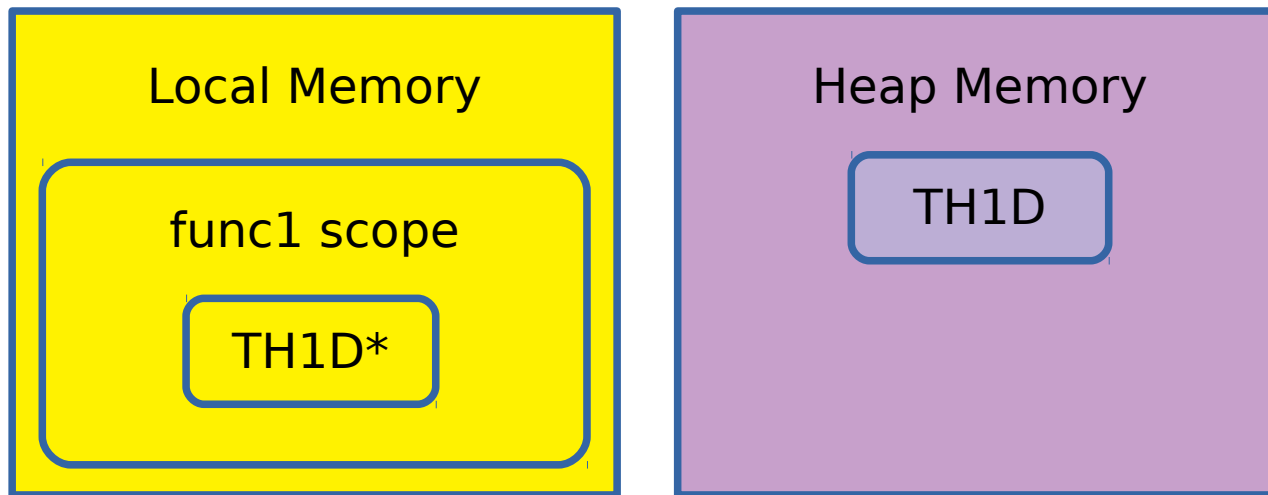
```
▶ void func1(){  
    TH1D *hp = new TH1D("h1","h1",100,-5,5);  
    hp->FillRandom("gaus",1000);  
    hp->Draw();  
}
```



Utilising The Heap

```
void func1(){
```

```
    ▶ TH1D *hp = new TH1D("h1","h1",100,-5,5);  
    hp->FillRandom("gaus",1000);  
    hp->Draw();  
}
```



TH1D object created on heap by new

Utilising The Heap

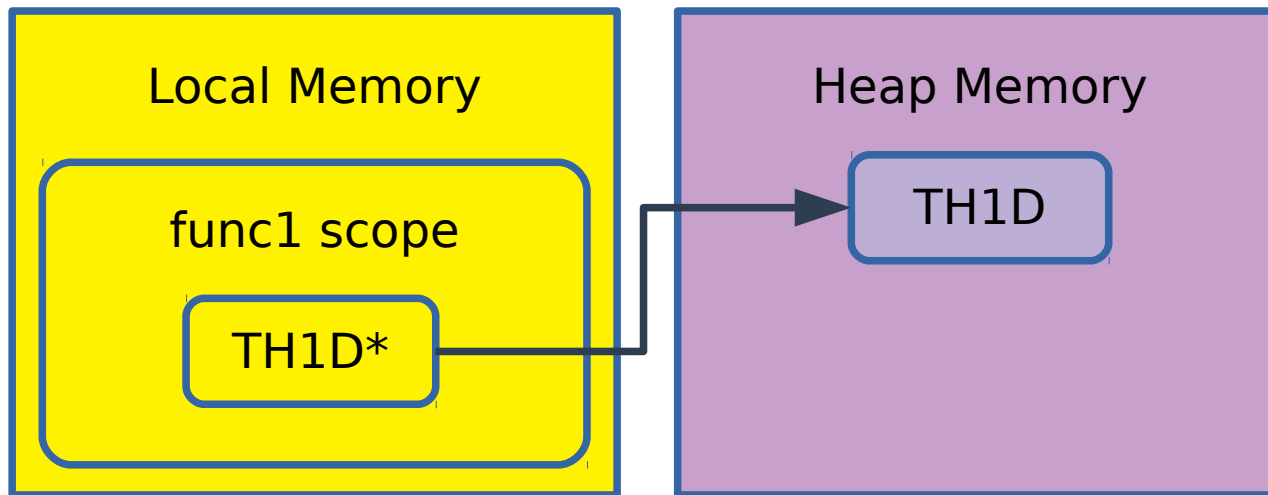
```
void func1(){
```

```
    TH1D *hp = new TH1D("h1","h1",100,-5,5);
```

```
    ▶ hp->FillRandom("gaus",1000);
```

```
    hp->Draw();
```

```
}
```



Pointer passes command to object

Utilising The Heap

```
void func1(){
```

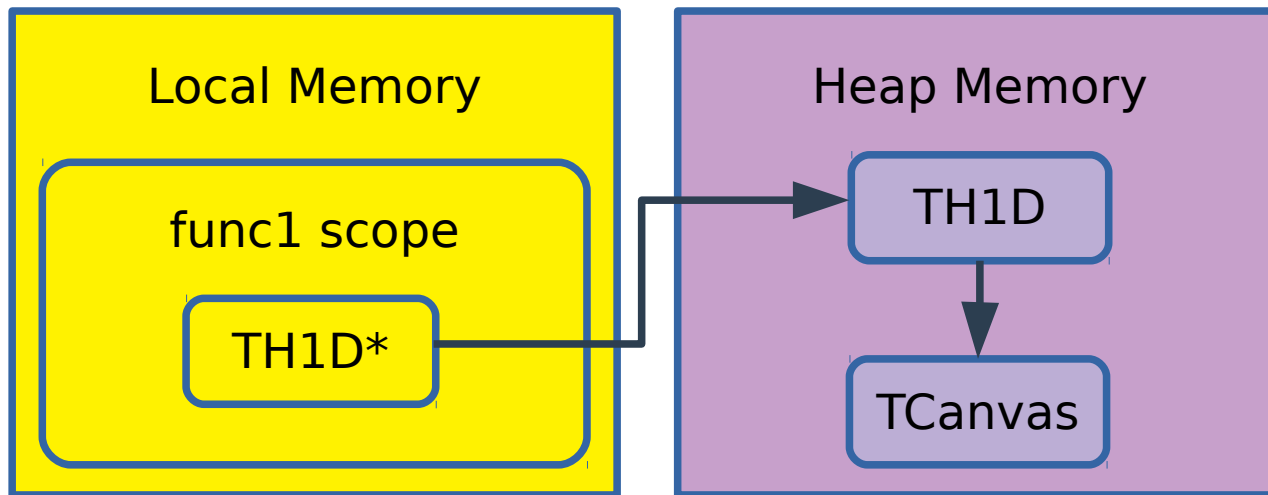
```
    TH1D *hp = new TH1D("h1","h1",100,-5,5);
```

```
    hp->FillRandom("gaus",1000);
```



```
    hp->Draw();
```

```
}
```



Draw to current pad (TCanvas created if needed)

Utilising The Heap

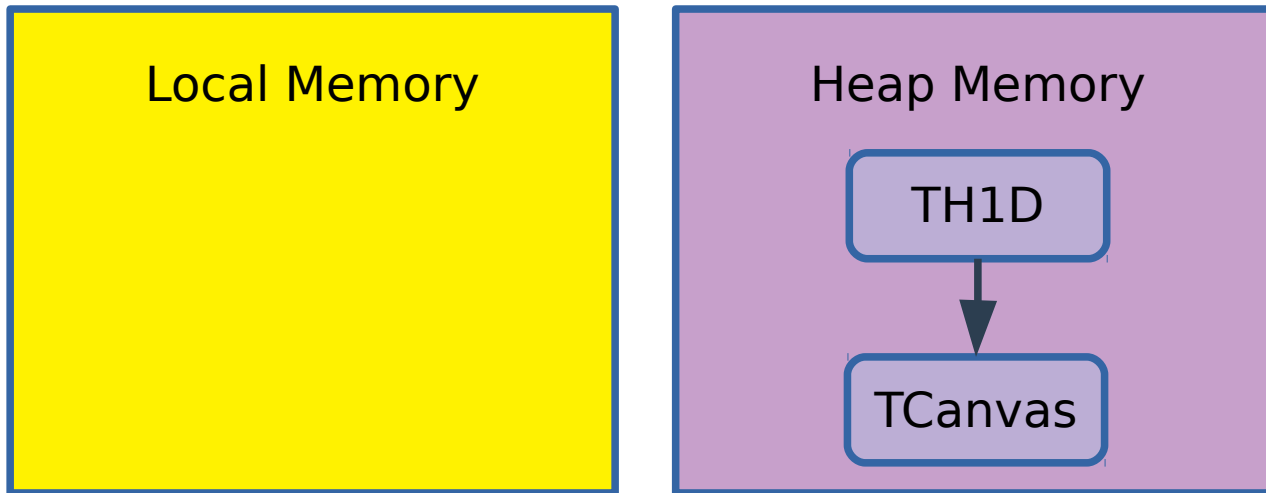
```
void func1(){
```

```
    TH1D *hp = new TH1D("h1","h1",100,-5,5);
```

```
    hp->FillRandom("gaus",1000);
```

```
    hp->Draw();
```

```
}
```



func1 scope variables deleted
Heap objects remain in memory

Pointer Danger

ROOT predominantly uses heap memory

**Pointers hold a memory address, heap OR local
(objects created with new are on the heap)**

Many ROOT functions return an address/pointer

**Pointers unchanged by object deletion
continue to point to (empty) address**

“Ownership” is important