Tricky Declarations

Kernighan

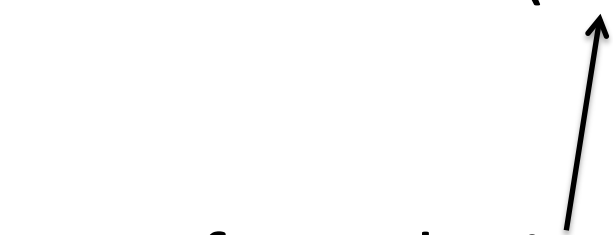
- “If I could change something about C …”

Remember two things

1. Start from the inside out (from the *var* out)
2. Operator precedence order
   - [ ]
   - ( )
   - *

Example

char (*v)[20];



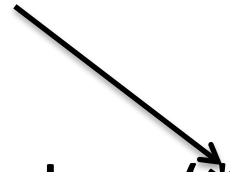
start from the inside

Cheat sheet. order:

- [ ]
- ( )
- *

Example

it’s a pointer



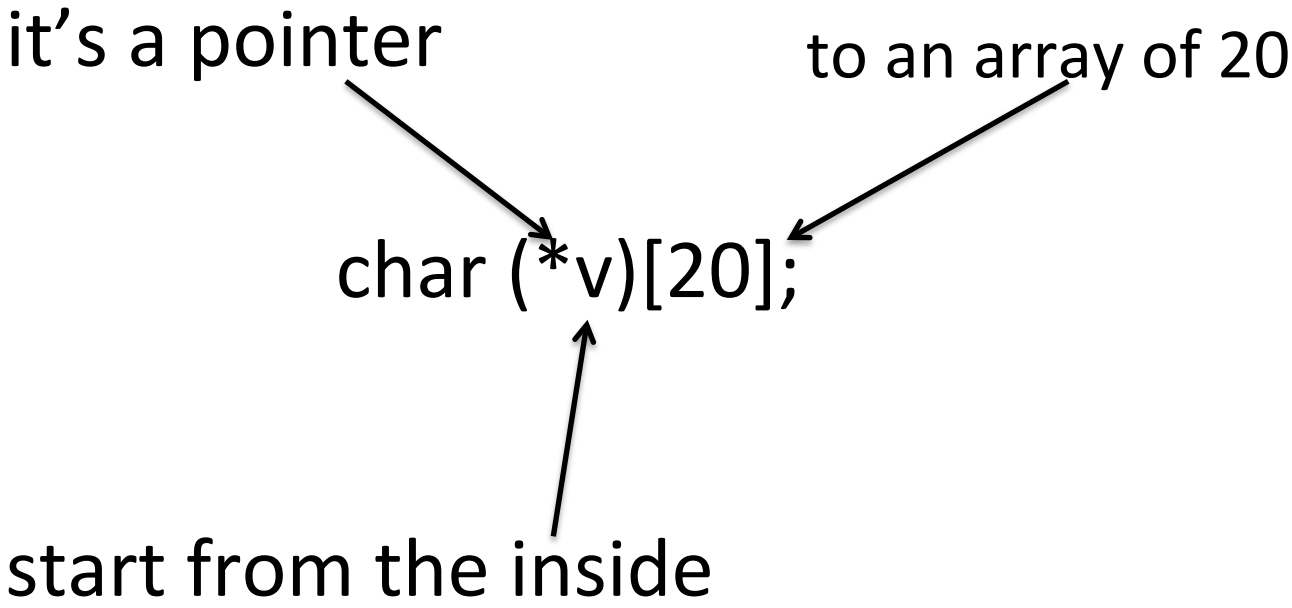
```
char (*v)[20];
```

start from the inside

Cheat sheet. order:

- [ ]
- ( )
- *

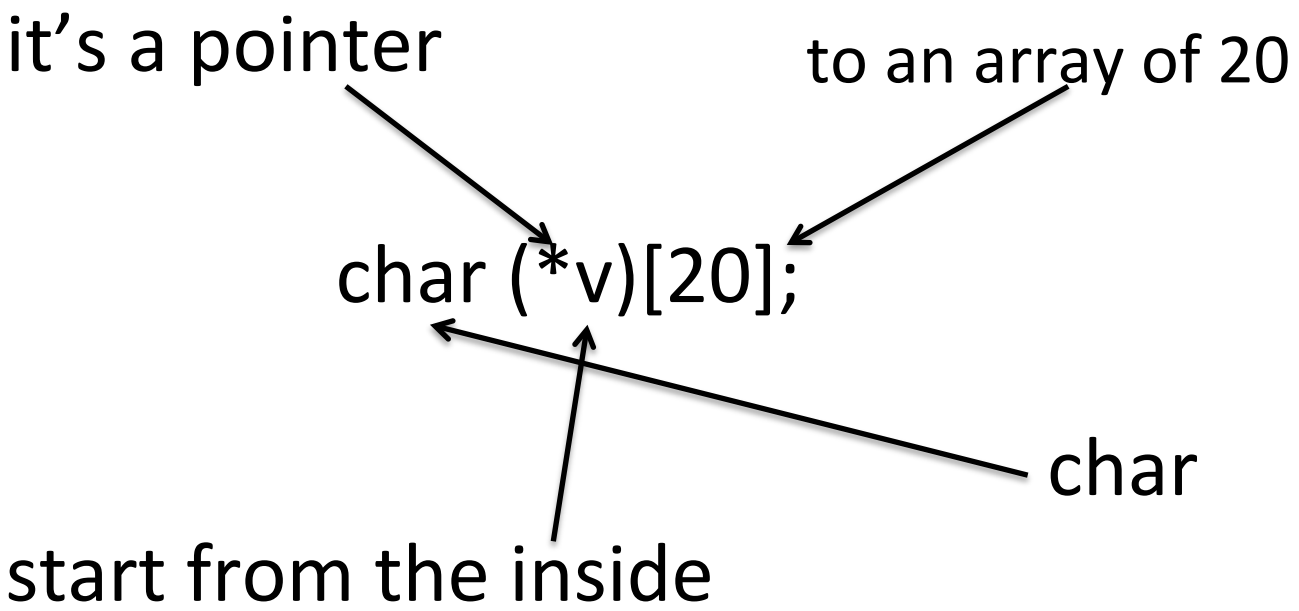
Example



Cheat sheet. order:

- [ ]
- ( )
- *

Example



Cheat sheet. order:

- [ ]
- ( )
- *

int *v[3]

int *v[3]

array of 3 pointers to int

int *(v[3])

int *(v[3])

array of 3 pointers to int

int (*v)[3]

int (*v)[3]

pointer to an array of 3 ints

int *v( )

int *v( )

function returning a pointer to int

```
void(*v)(int);
```

```
void(*v)(int);
```

pointer to a function that takes an int argument and returns nothing

```
int(*v)(void);
```

```
int(*v)(void);
```

pointer to a function that takes no argument and returns an int

int(*v[])()

int(*v[])()

array of pointers to functions returning int

int*(*v[10])()

int*(*v[10])()

array of 10 pointers to functions which take no arguments and return an int pointer

```
int (*(*f())[13])()
```

```
int (*(*f())[13])()
```

a function returning a pointer to an array of 13 pointers to functions returning int

Avoiding these

- Use typedef
- Recall:
  - typedef int bool;
- Instead of int (*(*x[3])())[5] :

```
typedef int fiveints[5];
typedef fiveints* p5i;
typedef p5i (*f_of_p5is)( );
f_of_p5is x[3];
```

- x is an array of 3 elements, each of which is a pointer to a function returning an array of 5 ints