*A little bit of typing relief…*

# Typedef

# Typedefs

- Typedef is used to create a name for data types.
- It is used to clarify types in your code (give a name to something that is commonly used) and to avoid entering long type definitions.
- It is most useful with structures.

# Typedef Example

```
struct data {
    int count;
    char name[30];
};
typedef struct data info;
```

new type name

structure being typedef’d

# Typedef Example

- Now you can use `info` anywhere in your program where `struct data` would have been placed.
- E.g. Define a structure and pointer to type info.

```
info record;  /*same as struct data record;*/

info *ptr;  /* same as struct data *ptr; */
```

# Another Use

- Another use for `typedef` is to create a *pointer* type.

  ```
  typedef struct data* infoPtr;
  ```

  - this is a pointer to the type `struct data`

- This can be confusing if it is not clear that the pointer is part of the `typedef`.

# Alternate Syntax

- Another syntax for typedef is to include it in the `struct` definition.

- Same as the previous `typedef` but combines two operations - declaring the `struct` and creating a type.

```
typedef struct {
    int count;
    char name[30];
} info;
```