4. Jumping from IPv4 to IPv6

But I just want to know what to change in my code to get it going with IPv6! Tell me now!
Ok! Ok!
Almost everything in here is something I've gone over, above, but it's the short version for the impatient. (Of course, there is more than this, but this is what applies to the guide.)

1. First of all, try to use **`getaddrinfo()`** to get all the `struct sockaddr` info, instead of packing the structures by hand. This will keep you IP version-agnostic, and will eliminate many of the subsequent steps.
2. Any place that you find you're hard-coding anything related to the IP version, try to wrap up in a helper function.
3. Change `AF_INET` to `AF_INET6`.
4. Change `PF_INET` to `PF_INET6`.
5. Change `INADDR_ANY` assignments to `in6addr_any` assignments, which are slightly different:

   ```
   struct sockaddr_in sa;
   struct sockaddr_in6 sa6;

   sa.sin_addr.s_addr = INADDR_ANY;  // use my IPv4 address
   sa6.sin6_addr = in6addr_any; // use my IPv6 address
   ```

   Also, the value `IN6ADDR_ANY_INIT` can be used as an initializer when the `struct in6_addr` is declared, like so:

   ```
   struct in6_addr ia6 = IN6ADDR_ANY_INIT;
   ```

6. Instead of `struct sockaddr_in` use `struct sockaddr_in6`, being sure to add "6" to the fields as appropriate (see `struct`s, above). There is no *`sin6_zero`* field.
7. Instead of `struct in_addr` use `struct in6_addr`, being sure to add "6" to the fields as appropriate (see `struct`s, above).
8. Instead of **`inet_aton()`** or **`inet_addr()`**, use **`inet_pton()`**.
9. Instead of **`inet_ntoa()`**, use **`inet_ntop()`**.
10. Instead of **`gethostbyname()`**, use the superior **`getaddrinfo()`**.
11. Instead of **`gethostbyaddr()`**, use the superior **`getnameinfo()`** (although **`gethostbyaddr()`** can still work with IPv6).
12. `INADDR_BROADCAST` no longer works. Use IPv6 multicast instead.

Et voila!