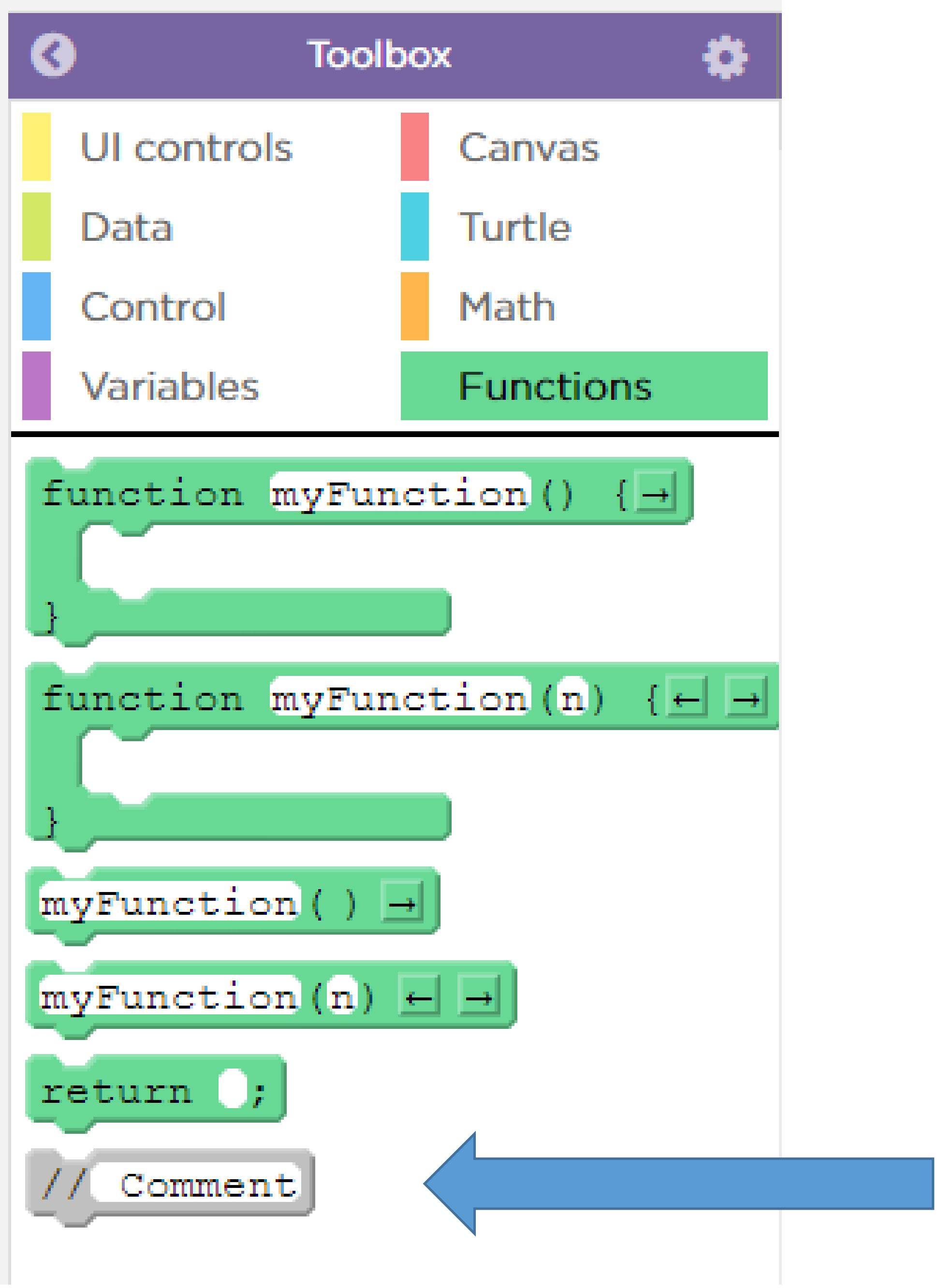
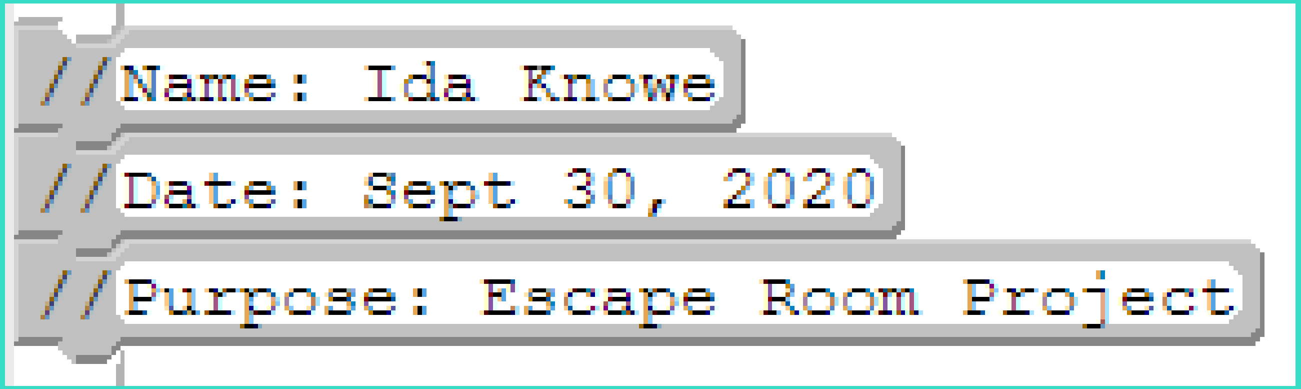
Commenting and Indenting



```
function myFunction() {→
}
function myFunction(n) {← →
}
myFunction( ) →
myFunction(n) ← →
return ;
// Comment
```

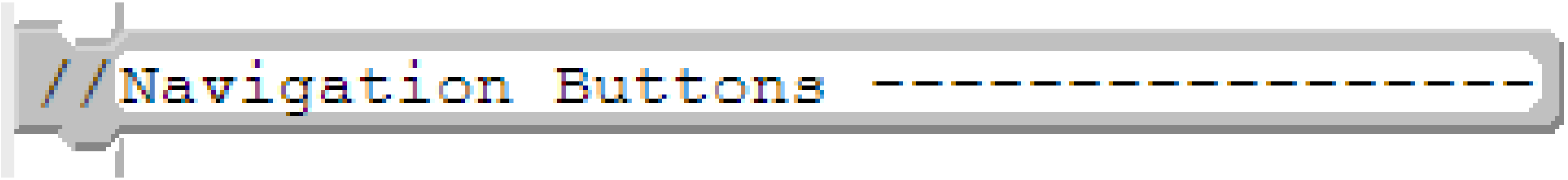
A comment is code that doesn't run.

This seems like it should be useless, but it really isn't.



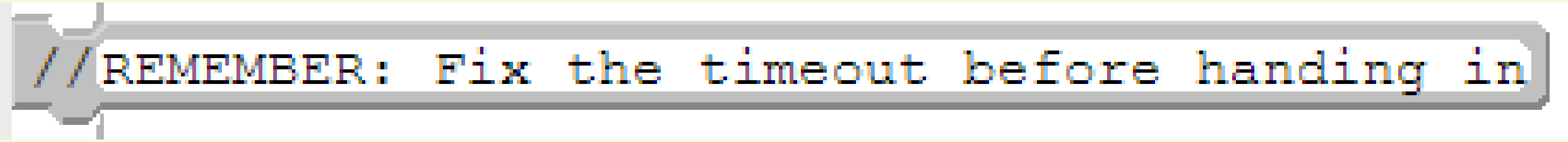
```
//Name: Ida Knowe
//Date: Sept 30, 2020
//Purpose: Escape Room Project
```

- Title comments – Name, Date and Purpose
- Appear at the top of the code
- Are used to “sign” your work.



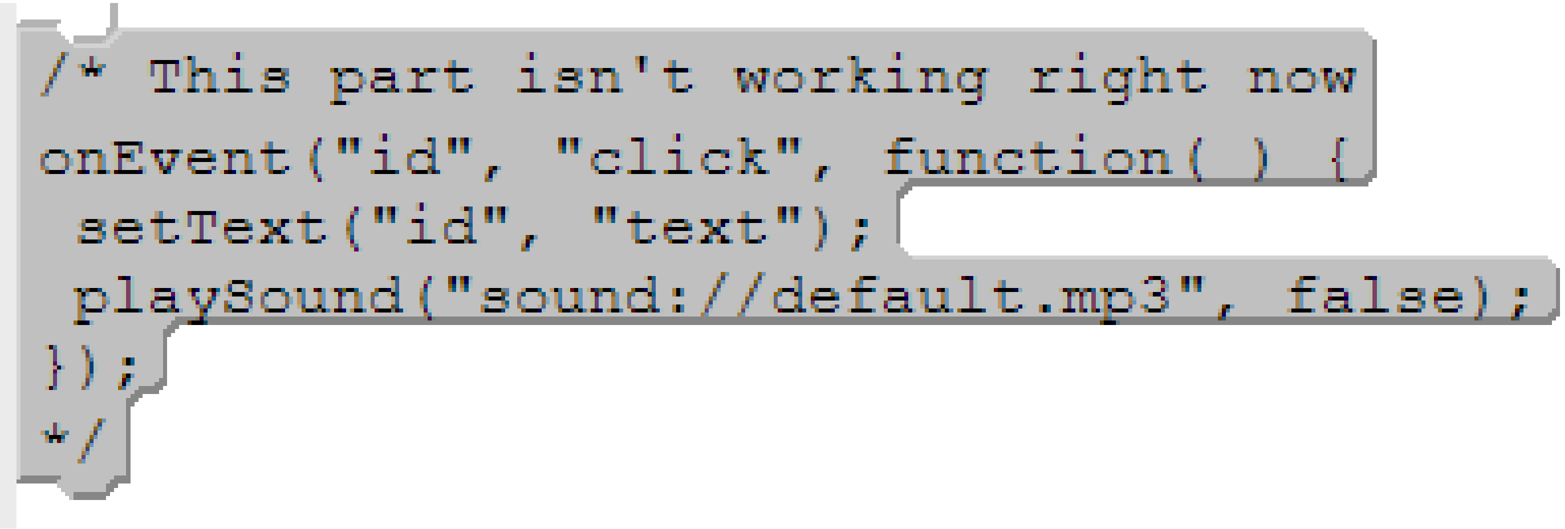
```
//Navigation Buttons -----------------
```

- Subtitles in the code
- We can use ------ or other symbols to separate and organize our code.



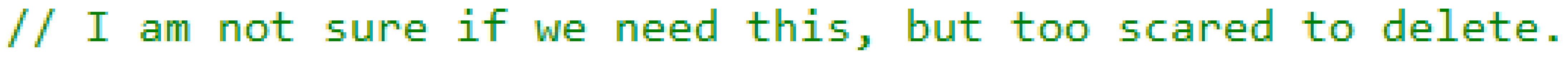
```
//REMEMBER: Fix the timeout before handing in
```

- Notes to yourself
- Sometimes it is handy to leave yourself a note

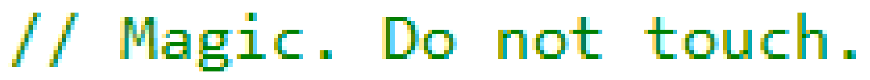


```
/* This part isn't working right now
onEvent("id", "click", function( ) {
  setText("id", "text");
  playSound("sound://default.mp3", false);
});
*/
```

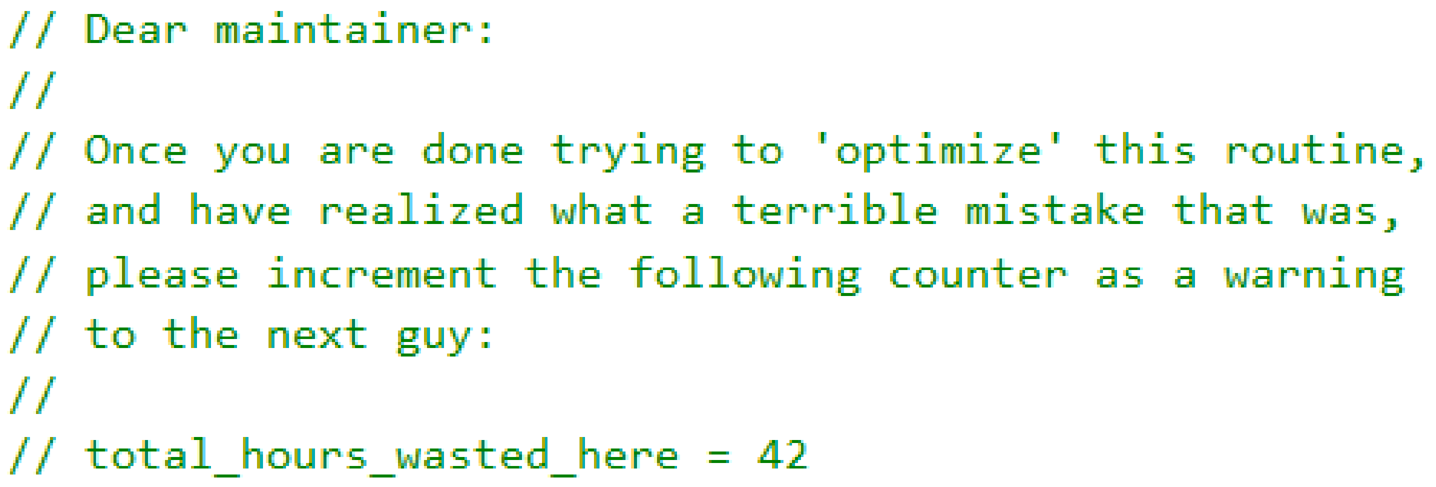
- Commenting out code
- Use /* and */ to temporarily take out code
- Lets you save code that isn’t running, but still run the code to test other things.



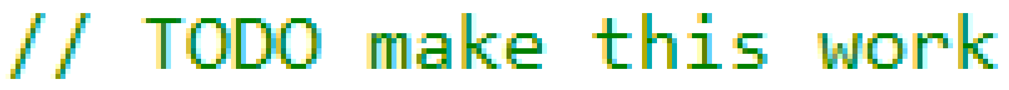
```
// I am not sure if we need this, but too scared to delete.
```



```
// Magic. Do not touch.
```

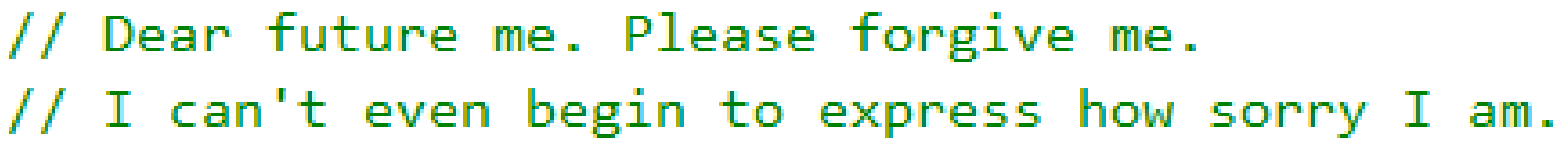


```
// Dear maintainer:
//
// Once you are done trying to 'optimize' this routine,
// and have realized what a terrible mistake that was,
// please increment the following counter as a warning
// to the next guy:
//
// total_hours_wasted_here = 42
```

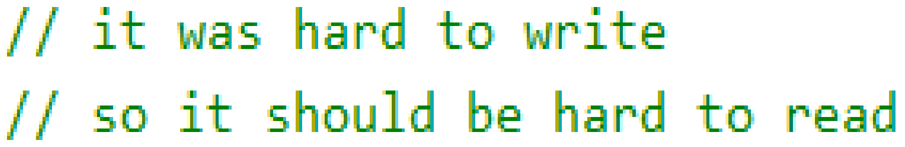


```
// TODO make this work
```

Some silly comments put in code by programmers

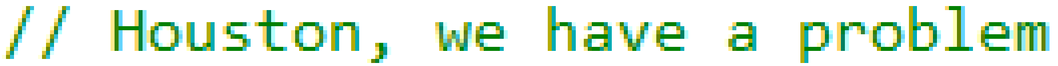


```
// Dear future me. Please forgive me.
// I can't even begin to express how sorry I am.
```

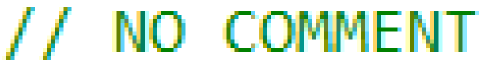


```
// it was hard to write
// so it should be hard to read
```

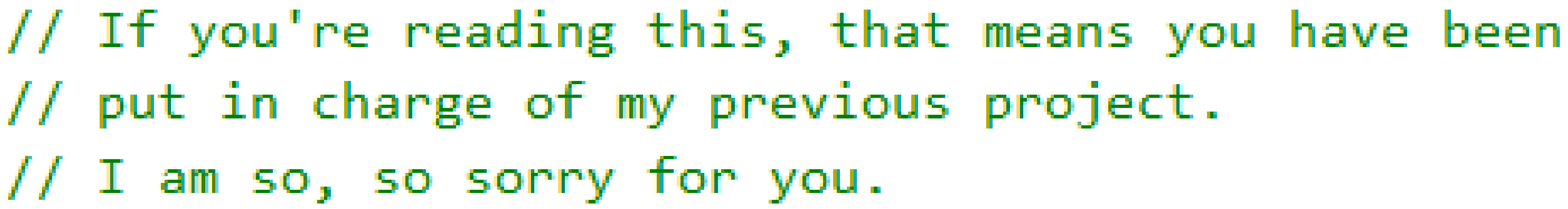
More silly comments



```
// Houston, we have a problem
```



```
// NO COMMENT
```



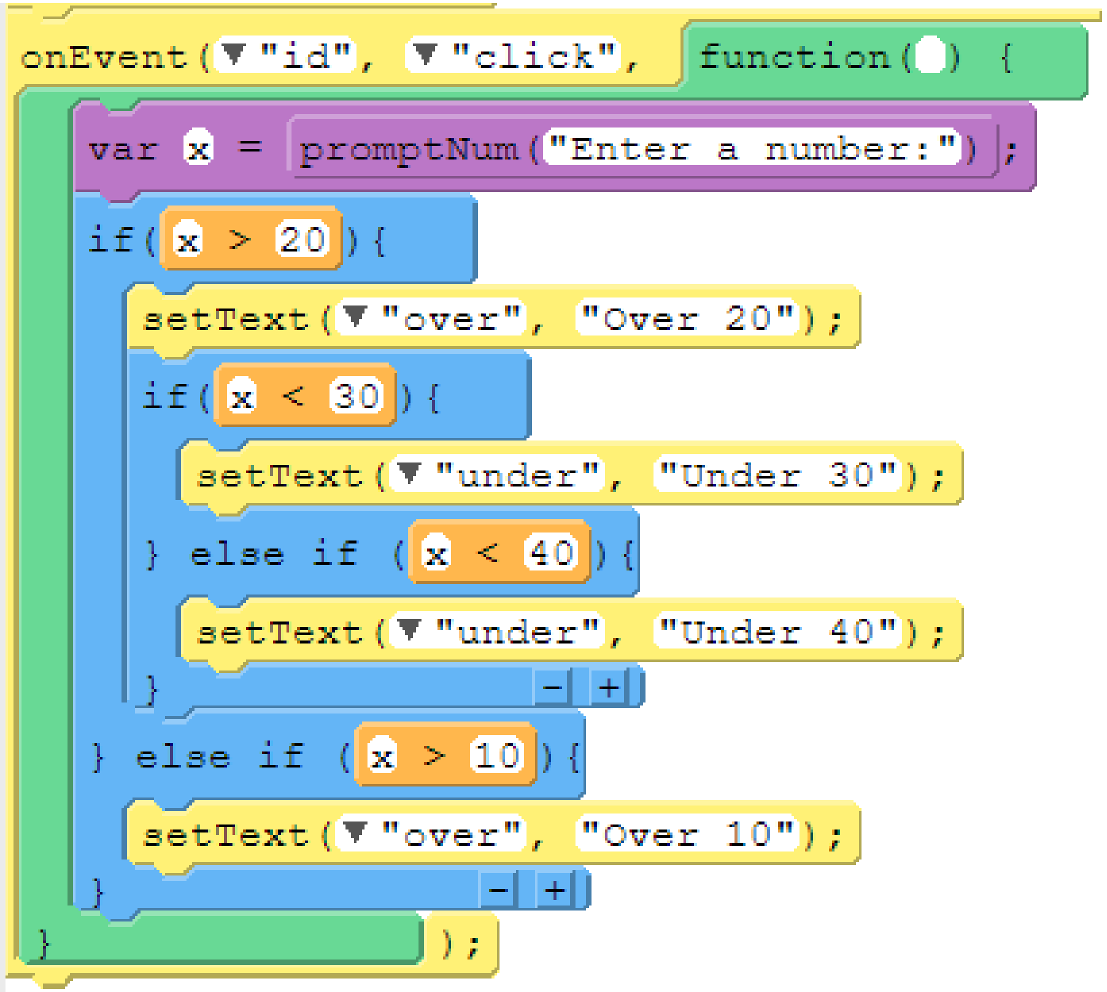
```
// If you're reading this, that means you have been
// put in charge of my previous project.
// I am so, so sorry for you.
```

Indenting your code is like sentences and paragraphs in English.

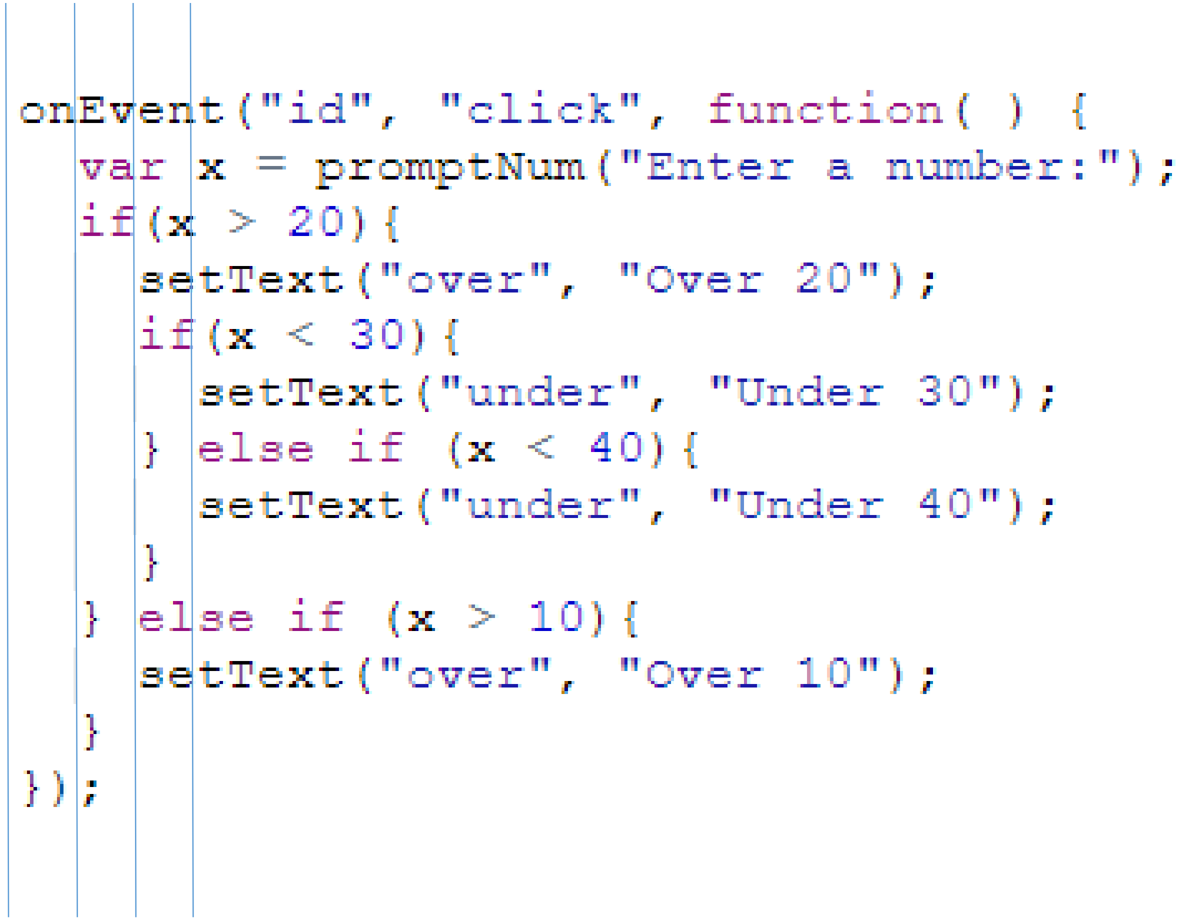
It makes your code easier to read and allows others to understand it.

It also can make it easier for you to see where a section of the code starts and ends.

What does well indented code look like?

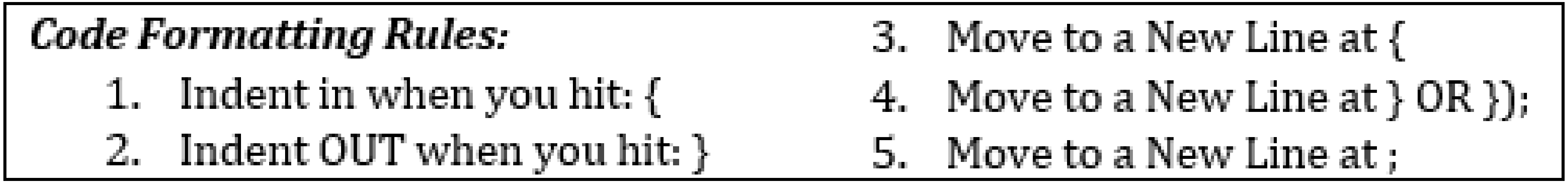


```
onEvent(▼"id", ▼"click", function( ) {
  var x = promptNum("Enter a number:");
  if(x > 20){
    setText(▼"over", "Over 20");
    if(x < 30){
      setText(▼"under", "Under 30");
    } else if (x < 40){
      setText(▼"under", "Under 40");
    }
  } else if (x > 10){
    setText(▼"over", "Over 10");
  }
});
```



```
onEvent("id", "click", function( ) {
  var x = promptNum("Enter a number:");
  if(x > 20){
    setText("over", "Over 20");
    if(x < 30){
      setText("under", "Under 30");
    } else if (x < 40){
      setText("under", "Under 40");
    }
  } else if (x > 10){
    setText("over", "Over 10");
  }
});
```

How do we indent code?



Code Formatting Rules:

1. Indent in when you hit: {
2. Indent OUT when you hit: }
3. Move to a New Line at {
4. Move to a New Line at } OR });
5. Move to a New Line at ;

```
onEvent("AngryBirds", "click", function( ) { var lvl =
prompt("Level? ");  if (lvl > 17) {   showElement("Blues");  }
else {  hideElement("Blues");  }   });
```

Code Formatting Rules:

1. Indent in when you hit: {
2. Indent OUT when you hit: }
3. Move to a New Line at {
4. Move to a New Line at } OR });
5. Move to a New Line at ;

```
onEvent("AngryBirds", "click", function( ) { var lvl =
prompt("Level? ");  if (lvl > 17) {   showElement("Blues");  }
else {  hideElement("Blues");  }   });
```

Code Formatting Rules:

1. Indent in when you hit: {
2. Indent OUT when you hit: }
3. Move to a New Line at {
4. Move to a New Line at } OR });
5. Move to a New Line at ;

```
onEvent("AngryBirds", "click", function( ) {
```

```
onEvent("AngryBirds", "click", function( ) { var lvl =
prompt("Level? ");  if (lvl > 17) {   showElement("Blues");  }
else {  hideElement("Blues");  }   });
```

Code Formatting Rules:

1. Indent in when you hit: {
2. Indent OUT when you hit: }
3. Move to a New Line at {
4. Move to a New Line at } OR });
5. Move to a New Line at ;

```
onEvent("AngryBirds", "click", function( ) {
    var lvl = prompt("Level? ");
```

```
onEvent("AngryBirds", "click", function( ) { var lvl =
prompt("Level? "); if (lvl > 17) {  showElement("Blues"); }
else {  hideElement("Blues"); }  });
```

Code Formatting Rules:

1. Indent in when you hit: {
2. Indent OUT when you hit: }
3. Move to a New Line at {
4. Move to a New Line at } OR });
5. Move to a New Line at ;

```
onEvent("AngryBirds", "click", function( ) {
    var lvl = prompt("Level? ");
    if (lvl > 17) {
```

```
onEvent("AngryBirds", "click", function( ) { var lvl =
prompt("Level? ");  if (lvl > 17) {   showElement("Blues");  }
else {  hideElement("Blues");  }   });
```

Code Formatting Rules:

1. Indent in when you hit: {
2. Indent OUT when you hit: }
3. Move to a New Line at {
4. Move to a New Line at } OR });
5. Move to a New Line at ;

```
onEvent("AngryBirds", "click", function( ) {
    var lvl = prompt("Level? ");
    if (lvl > 17) {
        showElement("Blues");
```

```
onEvent("AngryBirds", "click", function( ) { var lvl =
prompt("Level? ");  if (lvl > 17) {   showElement("Blues");  }
else {  hideElement("Blues");  }   });
```

Code Formatting Rules:

1. Indent in when you hit: {
2. Indent OUT when you hit: }
3. Move to a New Line at {
4. Move to a New Line at } OR });
5. Move to a New Line at ;

```
onEvent("AngryBirds", "click", function( ) {
    var lvl = prompt("Level? ");
    if (lvl > 17) {
        showElement("Blues");
    }
```

```
onEvent("AngryBirds", "click", function( ) { var lvl =
prompt("Level? "); if (lvl > 17) {  showElement("Blues"); }
else {  hideElement("Blues"); }  });
```

Code Formatting Rules:

1. Indent in when you hit: {
2. Indent OUT when you hit: }
3. Move to a New Line at {
4. Move to a New Line at } OR });
5. Move to a New Line at ;

```
onEvent("AngryBirds", "click", function( ) {
    var lvl = prompt("Level? ");
    if (lvl > 17) {
        showElement("Blues");
    }
    else {
```

```
onEvent("AngryBirds", "click", function( ) { var lvl =
prompt("Level? ");  if (lvl > 17) {   showElement("Blues");  }
else {  hideElement("Blues");  }   });
```

Code Formatting Rules:

1. Indent in when you hit: {
2. Indent OUT when you hit: }
3. Move to a New Line at {
4. Move to a New Line at } OR });
5. Move to a New Line at ;

```
onEvent("AngryBirds", "click", function( ) {
    var lvl = prompt("Level? ");
    if (lvl > 17) {
        showElement("Blues");
    }
    else {
        hideElement("Blues");
```

```
onEvent("AngryBirds", "click", function( ) { var lvl =
prompt("Level? ");  if (lvl > 17) {   showElement("Blues");  }
else {  hideElement("Blues");  }   });
```

Code Formatting Rules:

1. Indent in when you hit: {
2. Indent OUT when you hit: }
3. Move to a New Line at {
4. Move to a New Line at } OR });
5. Move to a New Line at ;

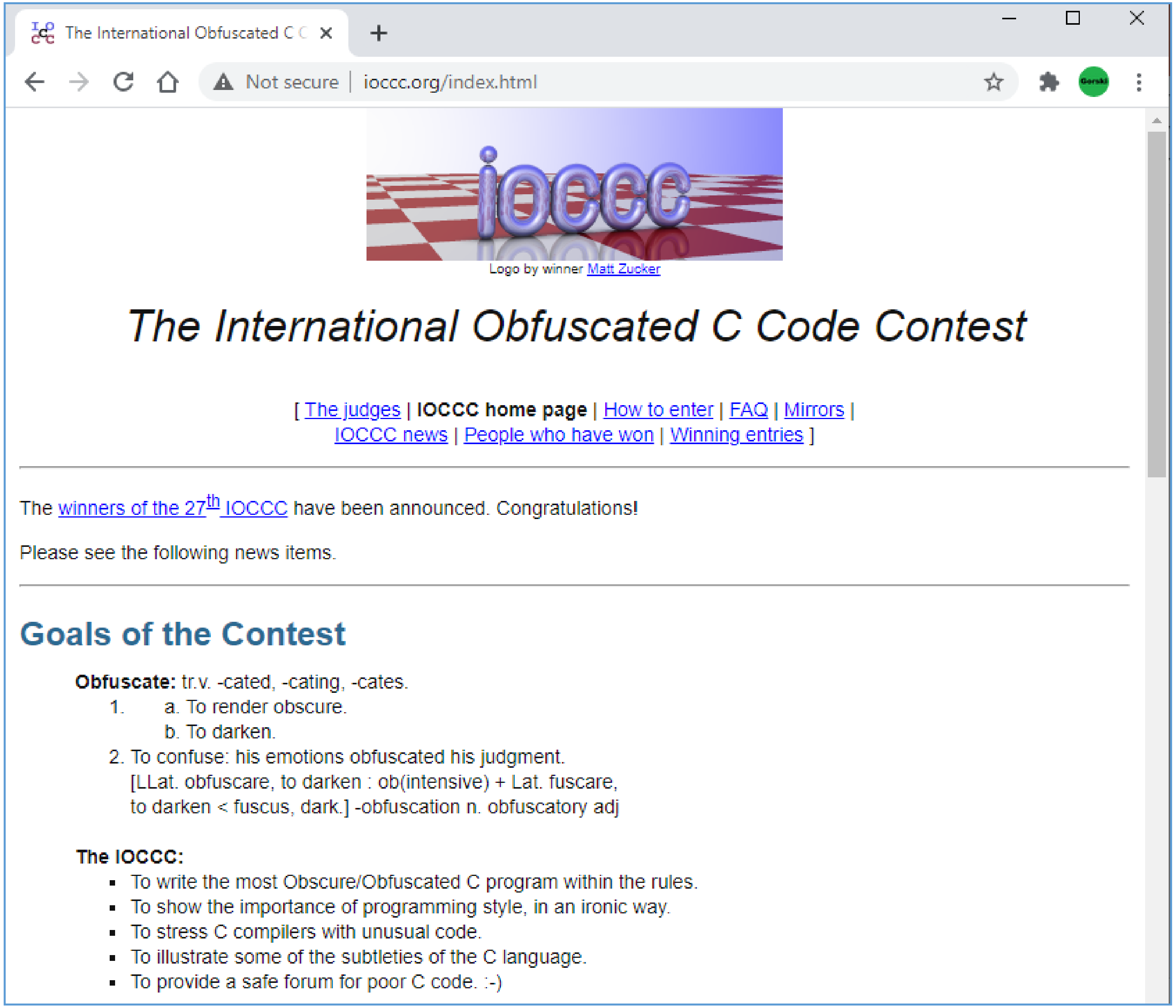
```
onEvent("AngryBirds", "click", function( ) {
    var lvl = prompt("Level? ");
    if (lvl > 17) {
        showElement("Blues");
    }
    else {
        hideElement("Blues");
    }
```

```
onEvent("AngryBirds", "click", function( ) { var lvl =
prompt("Level? "); if (lvl > 17) {  showElement("Blues"); }
else {  hideElement("Blues"); }  });
```

Code Formatting Rules:

1. Indent in when you hit: {
2. Indent OUT when you hit: }
3. Move to a New Line at {
4. Move to a New Line at } OR });
5. Move to a New Line at ;

```
onEvent("AngryBirds", "click", function( ) {
    var lvl = prompt("Level? ");
    if (lvl > 17) {
        showElement("Blues");
    }
    else {
        hideElement("Blues");
    }
});
```



The International Obfuscated C C
Not secure | ioccc.org/index.html

Logo by winner Matt Zucker

The International Obfuscated C Code Contest

[The judges | **IOCCC home page** | How to enter | FAQ | Mirrors |
IOCCC news | People who have won | Winning entries]

The winners of the 27th IOCCC have been announced. Congratulations!

Please see the following news items.

Goals of the Contest

Obfuscate: tr.v. -cated, -cating, -cates.

1. a. To render obscure.
   b. To darken.
2. To confuse: his emotions obfuscated his judgment.
   [LLat. obfuscare, to darken : ob(intensive) + Lat. fuscare,
   to darken < fuscus, dark.] -obfuscation n. obfuscatory adj

The IOCCC:

- To write the most Obscure/Obfuscated C program within the rules.
- To show the importance of programming style, in an ironic way.
- To stress C compilers with unusual code.
- To illustrate some of the subtleties of the C language.
- To provide a safe forum for poor C code. :-)