



The Art of Readable Code

Part III: Reorganizing Your Code



Agenda

1. Extracting Unrelated Subproblem
2. One Task at a Time
3. Turning Thoughts into Code
4. Writing Less Code



Extracting Unrelated Subproblem

1. Introductory Example: `findClosestLocation()`
2. Pure Utility Code
3. Project-Specific Functionality
4. Simplifying an Existing Interface



Extracting Unrelated Subproblem

This principle is to makes code more robust and easier to read.



1.Introductory Example: findClosestLocation()

```
// Return which element of 'array' is closest to the given latitude/longitude.
// Models the Earth as a perfect sphere.
var findClosestLocation = function (lat, lng, array) {
  var closest;
  var closest_dist = Number.MAX_VALUE;
  for (var i = 0; i < array.length; i += 1) {
    // Convert both points to radians.
    var lat_rad = radians(lat);
    var lng_rad = radians(lng);
    var lat2_rad = radians(array[i].latitude);
    var lng2_rad = radians(array[i].longitude);

    // Use the "Spherical Law of Cosines" formula.
    var dist = Math.acos(Math.sin(lat_rad) * Math.sin(lat2_rad) +
                        Math.cos(lat_rad) * Math.cos(lat2_rad) *
                        Math.cos(lng2_rad - lng_rad));

    if (dist < closest_dist) {
      closest = array[i];
    }
  }
}
```



1.Introductory Example: findClosestLocation()

```
var spherical_distance = function (lat1, lng1, lat2, lng2) {  
  var lat1_rad = radians(lat1);  
  var lng1_rad = radians(lng1);  
  var lat2_rad = radians(lat2);  
  var lng2_rad = radians(lng2);  
  
  // Use the "Spherical Law of Cosines" formula.  
  return Math.acos(Math.sin(lat1_rad) * Math.sin(lat2_rad) +  
    Math.cos(lat1_rad) * Math.cos(lat2_rad) *  
    Math.cos(lng2_rad - lng1_rad));  
};
```

Now the remaining code becomes:

```
var findClosestLocation = function (lat, lng, array) {  
  var closest;  
  var closest_dist = Number.MAX_VALUE;  
  for (var i = 0; i < array.length; i += 1) {  
    var dist = spherical_distance(lat, lng, array[i].latitude, array[i].longitude);  
    if (dist < closest_dist) {  
      closest = array[i];  
      closest_dist = dist;  
    }  
  }  
  return closest;  
};
```



2. Pure Utility Code

Function read entire content of a file :

1. PHP :
 - a. `file_get_contents("filename")`
2. Python :
 - a. `open("filename").read()`

3. C++ :

```
ifstream file(file_name);

// Calculate the file's size, and allocate a buffer of that size.
file.seekg(0, ios::end);
const int file_size = file.tellg();
char* file_buf = new char [file_size];

// Read the entire file into the buffer.
file.seekg(0, ios::beg);
file.read(file_buf, file_size);
file.close();
```

3. Project-Specific Functionality

1. This code requires far less effort to read
2. The code should stay in the same file where it's used

```
business = Business()
business.name = request.POST["name"]

url_path_name = business.name.lower()
url_path_name = re.sub(r"['\.]", "", url_path_name)
url_path_name = re.sub(r"^[a-z0-9]+", "-", url_path_name)
url_path_name = url_path_name.strip("-")
business.url = "/biz/" + url_path_name

business.date_created = datetime.datetime.utcnow()
business.save_to_database()
```



```
CHARS_TO_REMOVE = re.compile(r"['\.]+")
CHARS_TO_DASH = re.compile(r"^[a-z0-9]+")

def make_url_friendly(text):
    text = text.lower()
    text = CHARS_TO_REMOVE.sub('', text)
    text = CHARS_TO_DASH.sub('-', text)
    return text.strip("-")
```

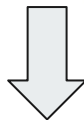


```
business = Business()
business.name = request.POST["name"]
business.url = "/biz/" + make_url_friendly(business.name)
business.date_created = datetime.datetime.utcnow()
business.save_to_database()
```


4. Simplifying an Existing Interface

1. **Library** mostly offers a clean **interface** if **not**, make your own “**wrapper**” functions.
2. For **example**, dealing with **browser** cookies in JavaScript is far from ideal.

```
var max_results;  
var cookies = document.cookie.split(';');  
for (var i = 0; i < cookies.length; i++) {  
    var c = cookies[i];  
    c = c.replace(/^[\ ]+/, ''); // remove leading spaces  
    if (c.indexOf("max_results=") === 0)  
        max_results = Number(c.substring(12, c.length));  
}
```

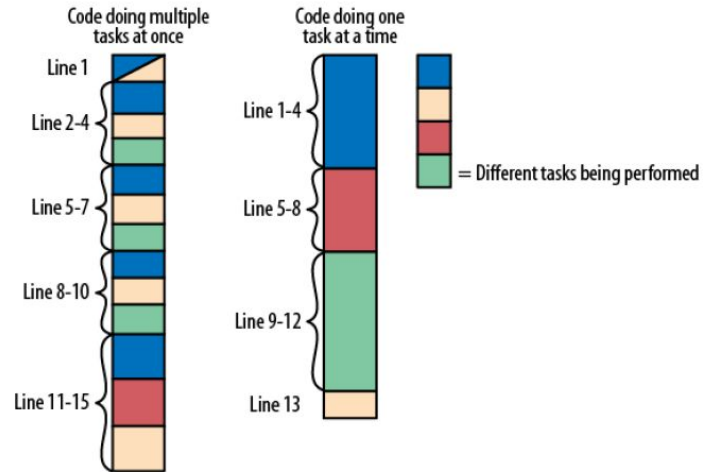


- `set_cookie(name, value, days_to_expire);`
- `delete_cookie(name);`

One Task at a Time

Tasks Can Be Small

- a. Code that does **multiple things** at once is harder to **understand**
- b. List out all the “tasks” your code is doing
- c. Try to separate those tasks as much as you can into different functions





1. Tasks Can Be Small

Suppose there's a voting widget on a blog where a user can vote a comment "Up" or "Down." The total score of a comment is the sum over all votes: +1 for each "Up" vote, -1 for each "Down" vote.

```
var vote_changed = function (old_vote, new_vote) {  
  var score = get_score();  
  
  if (new_vote !== old_vote) {  
    if (new_vote === 'Up') {  
      score += (old_vote === 'Down' ? 2 : 1);  
    } else if (new_vote === 'Down') {  
      score -= (old_vote === 'Up' ? 2 : 1);  
    } else if (new_vote === '') {  
      score += (old_vote === 'Up' ? -1 : 1);  
    }  
  }  
  
  set_score(score);  
};
```



1. Tasks Can Be Small

- solving each task separately

```
var vote_value = function (vote) {  
  if (vote === 'Up') {  
    return +1;  
  }  
  if (vote === 'Down') {  
    return -1;  
  }  
  return 0;  
};
```

Now the rest of the code can solve the second task, updating score:

```
var vote_changed = function (old_vote, new_vote) {  
  var score = get_score();  
  
  score -= vote_value(old_vote); // remove the old vote  
  score += vote_value(new_vote); // add the new vote  
  
  set_score(score);  
};
```



Turning Thoughts into Code

1. Describing Logic Clearly
2. Knowing Your Libraries Helps



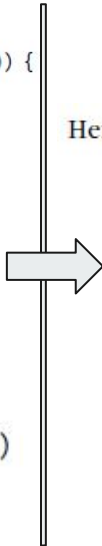
1. Describing Logic Clearly

```
$is_admin = is_admin_request();  
if ($document) {  
    if (!$is_admin && ($document['username'] != $_SESSION['username'])) {  
        return not_authorized();  
    }  
} else {  
    if (!$is_admin) {  
        return not_authorized();  
    }  
}
```

Let's start by describing the logic in plain English:

There are two ways you can be authorized:

- 1) you are an admin
 - 2) you own the current document (if there is one)
- Otherwise, you are not authorized.



Here is an alternative solution inspired by this description:

```
if (is_admin_request()) {  
    // authorized  
} elseif ($document && ($document['username'] == $_SESSION['username'])) {  
    // authorized  
} else {  
    return not_authorized();  
}
```



2. Knowing Your Libraries Helps

- What libraries should be used for improving the code
- Be aware of 'what your library has to offer.



Writing Less Code

1. Don't Bother Implementing That Feature—You Won't Need It
2. Keeping Your Codebase Small
3. Be Familiar with the Libraries Around You
 - a. Example: Lists and Sets in Python



Don't Bother Implementing That Feature- You Won't Need It

1. Implement only feature that are going to use
2. Don't overestimate how many feature truly needed to our project
3. Estimate how much effort it takes to implement a feature



2. Keeping Your Codebase Small

1. Remove duplicated code
2. Remove unused code or useless features
3. Keep your project compartmentalized into disconnected subprojects.
4. Be conscious of doing your code light weight



3. Be Familiar with the Libraries Around You

1. Capability understand the library code what you are using
2. Example: Lists and Sets in Python



Thank you