

The Principle of Least Surprise

“Write code, not mind games”

Intuitive Outcomes

Functions and classes should perform actions that another programmer can reasonably expect.

Converting a string to an enum.

```
Day day = DayDate.StringToName(String dayName);
```

StringToName should not be case dependent, and should accept common abbreviations.

“Tuesday” should have the same result as “tuesday” or “tues.”

Documentation Matches Implementation

If null is an option for a parameter don't throw an exception.

```
// Performs operation and stores
// optionalObject if not null.
Func(int x, int y, Object optionalObject) {
    other(x, y);
    if (optionalObject == null) {
        throw exception;
    }
    classObject = optionalObject;
}
```

```
// Performs operation and stores
// optionalObject if not null.
Func(int x, int y, Object optionalObject) {
    other(x, y);
    if (optionalObject != null) {
        classObject = optionalObject;
    }
}
```

Make Failures Obvious

Don't return success when something within the function fails.

```
StoreXValue(int x) {  
    Table valueTable;  
  
    valueTable = startTable();  
    if (valueTable == null) {  
        return SUCCESS;  
    }  
    valueTable.store(x);  
  
    return SUCCESS;  
}
```

```
StoreXValue(int x) {  
    Table valueTable;  
  
    valueTable = startTable();  
    if (valueTable == null) {  
        return FAILURE;  
    }  
    valueTable.store(x);  
  
    return SUCCESS;  
}
```