Before Adding any Scope to the project **🡺** the first question to ask is:

**What is the Cost of each solution and what are the benefits?**

When building a software **🡺**

**Do not model the universe! Only create enough to solve the problem at hand**

**Ctrl+D 🡺** Duplicates lines

**ViewModel 🡺** Is not part of the domain and purely for presentation. Therefore if you require to alter the domain model to only show specific i.e. DateTime object is the DomainModel and you only want to present the Date part of it.

**Usability Best Practices** – Reduce user’s stress and guess work. Scientifically proven that use Most Attention are the Top Left and Least Attention at the Right Bottom of the screen. This is for Left to Right readers like English.

**Implement Logical Delete Operation** – not physically deleted from DB, just mark it. This allows Undo and prevent multiple cascade problem.

# *Labels*

* Keep them short
* Be concise
* Don’t’ use ambiguous words
* Use **Top** or **Right** align

# *Fields*

* Reduce the number of fields
* Avoid optional fields
* Separate mandatory and optional fields
* Group related fields
* Specify the format
* Set focus on the first field

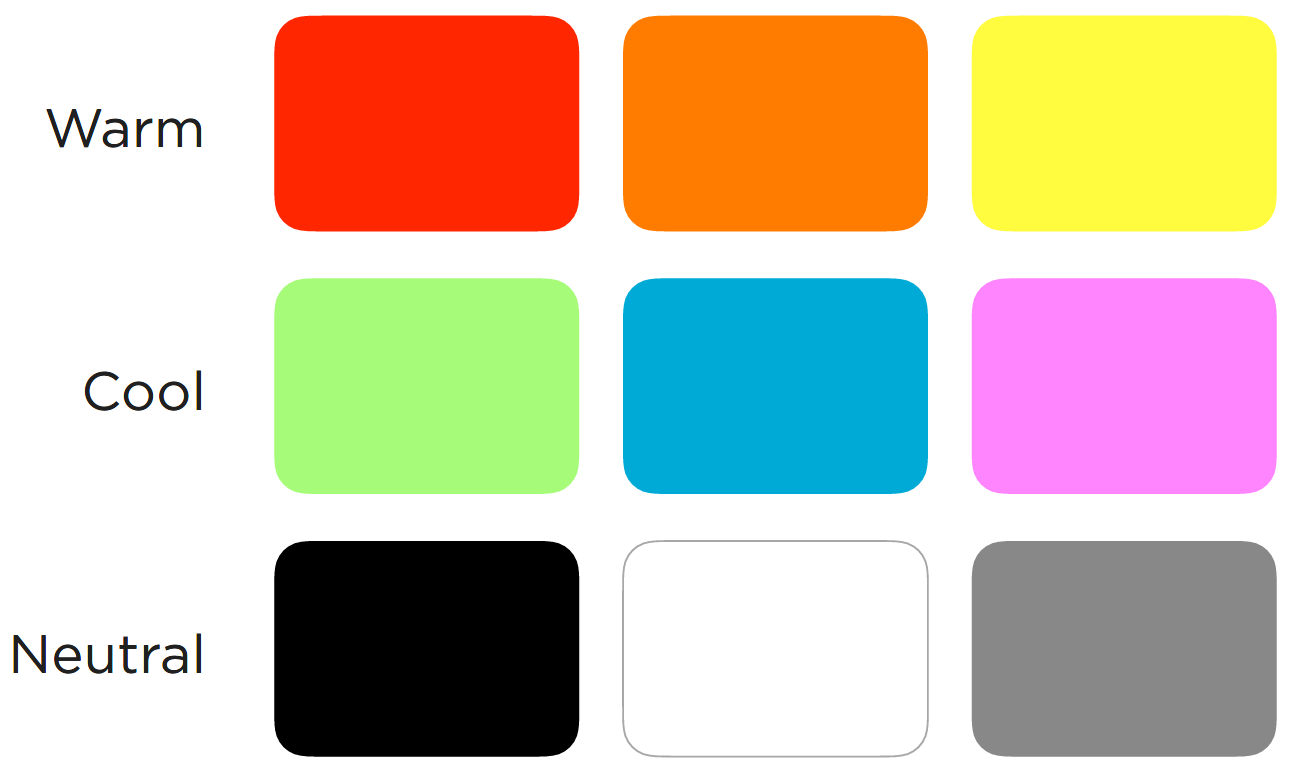
# *Actions*

* Each form should have a primary action
* Avoid secondary actions if possible
* Otherwise, visually separated them
* Align primary actions with input fields

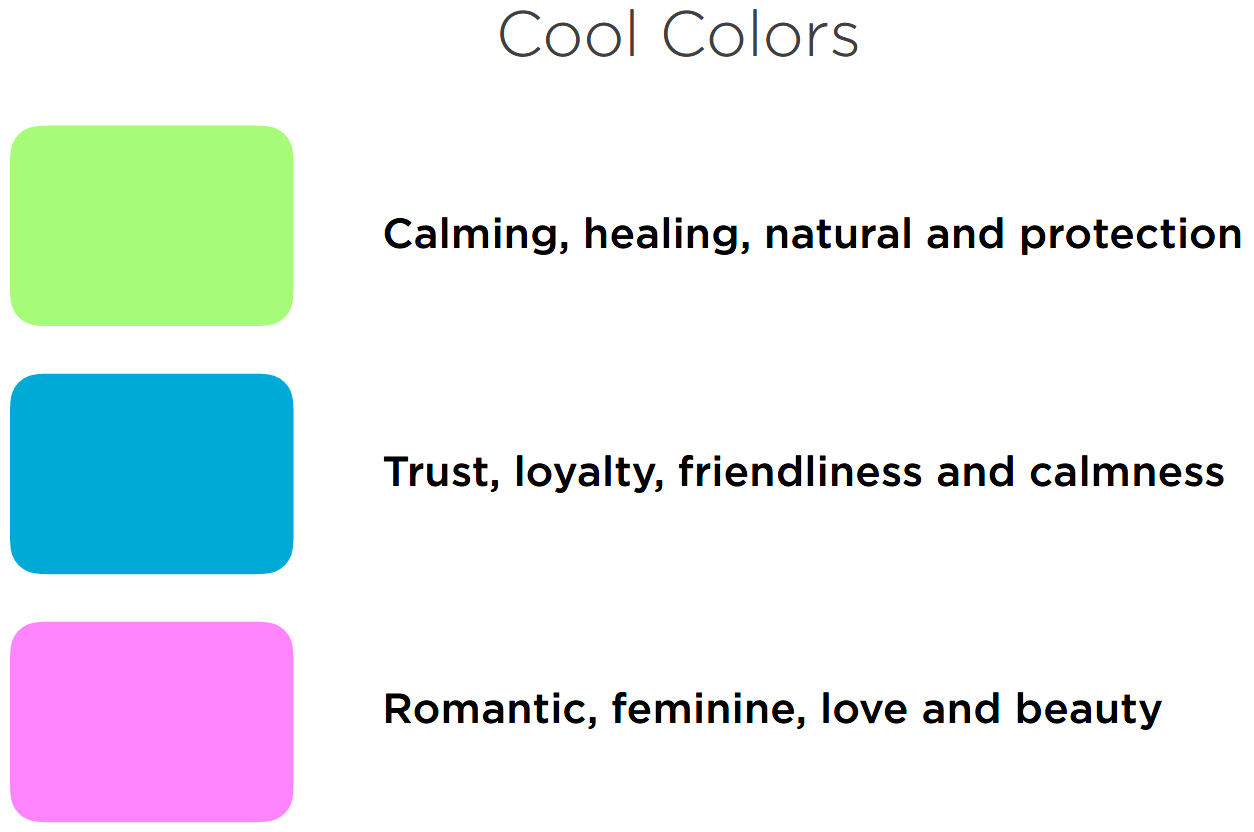
# *Validation*

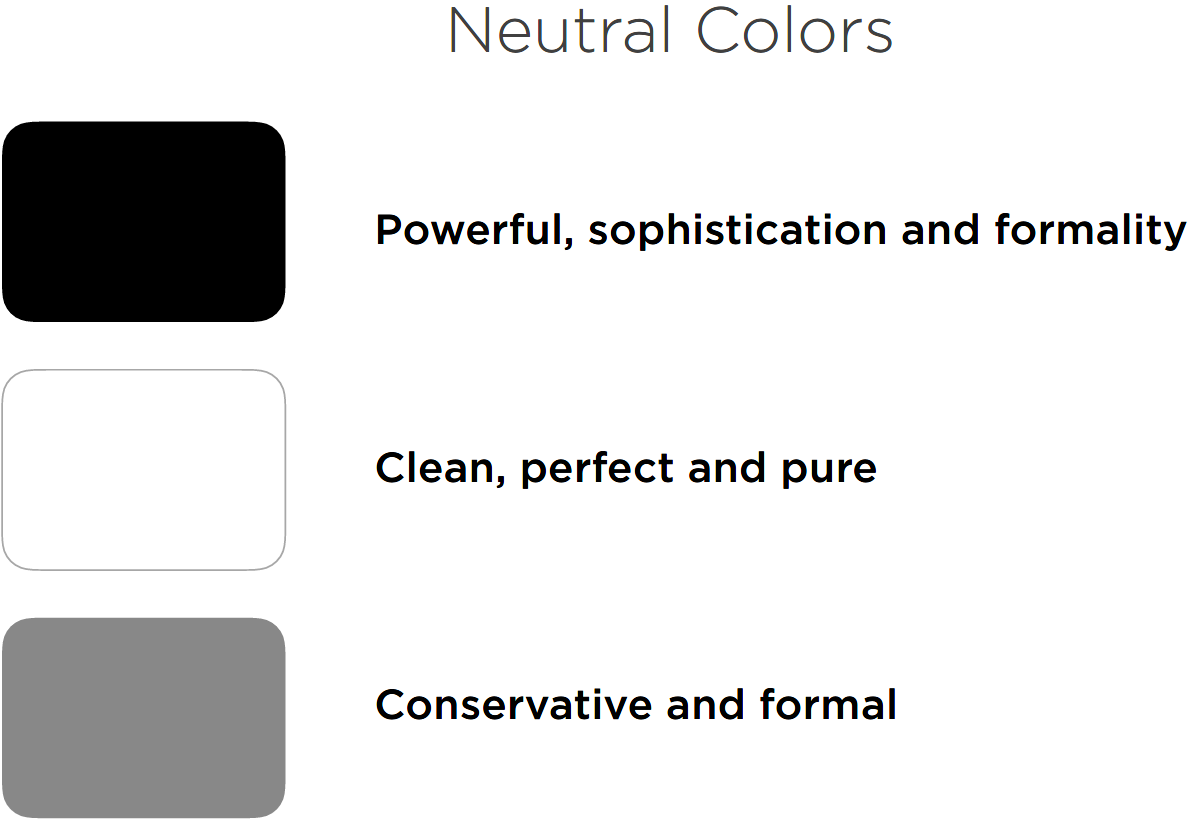
* Provide clear validation messages
* Use red to indicate errors
* Use green to indicate success
* Provide smart defaults

**Colour Implications**



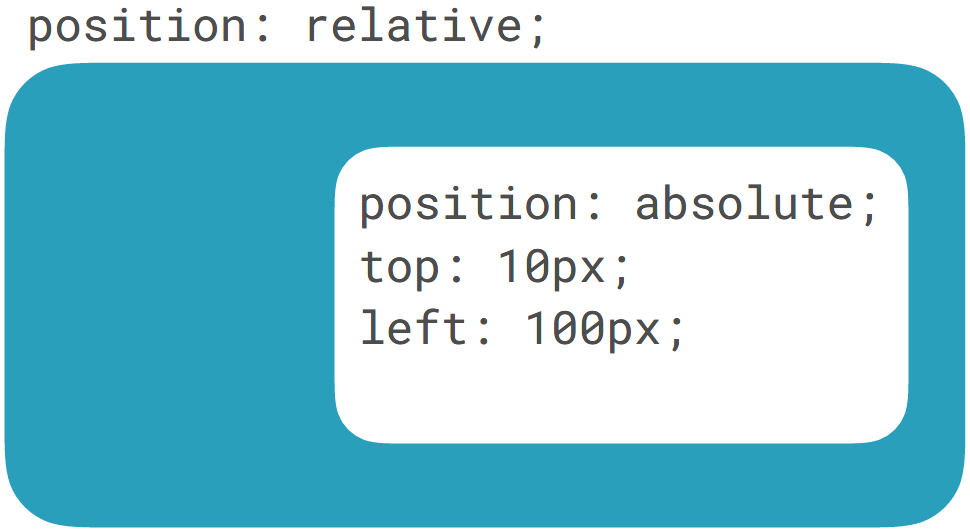






CSS Positioning

An element with **relative position** allows us to **absolutely** position its children.



Domain Model

**Persistence Ignorance** is an application of single responsibility principle, which in practice means that Domain Objects (DO) shouldn't contain code related to **persistence**, instead they should only contain domain logic.