

cond->>, cond->

Use when you might want to update

```
(defn record-vote [{:keys [age] :as m}]  
  (cond-> m (> age 20)  
    (assoc :voted? true)))
```

```
(record-vote {:name "Mark" :age 12})  
=> {:name "Mark", :age 12}
```

```
(record-vote {:name "Mike" :age 21})  
=> {:name "Mike", :age 21, :voted? true}
```

Another cond-> example

```
(defn normalize [{:keys [age date name] :as m}]
  (cond-> m
    (string? age) (update :age edn/read-string)
    (string? date) (update :date clojure.instant/read-instant-date)
    (not (re-matches #"[A-Z][a-z]+" name)) (update :name
                                                    (comp
                                                     cs/capitalize
                                                     cs/lower-case))))
```

```
(normalize
  {:name "sam"
   :age "54"
   :date "2018-08-11"})
=> {:name "Sam", :age 54, :date #inst"2018-08-11T00:00:00.000-00:00"}
```

```
(normalize
  {:name "Bob"
   :age 23
   :date #inst"2012-01-02"})
=> {:name "Bob", :age 23, :date #inst"2012-01-02T00:00:00.000-00:00"}
```