In[1]:= **Needs["RG`BaseUtils`"]**

In[2]:= **? RG`BaseUtils`***

▼ **RG`BaseUtils`**

assert	carryFirst	carryLast	reload

---

assert

In[3]:= **? assert**

assert[expr] evaluate expr with temporally enabled assertions

In[4]:= **Assert[1 > 2]**

Out[4]= Assert[1 > 2]

In[5]:= **assert[1 > 2]**

Assert::asrtfl : Assertion 1 > 2 at line 31 in RG`BaseUtils` failed. ≫

---

reload

In[6]:= **? reload**

reload[context] remove definitions context`* and reload it

In[7]:= **reload["RG`BaseUtils`", "verbose" → True]**

context: RG`BaseUtils`

list of symbols to remove: {assert, carryFirst, carryLast, reload}

list of loaded symbols: {assert, carryFirst, carryLast, reload}
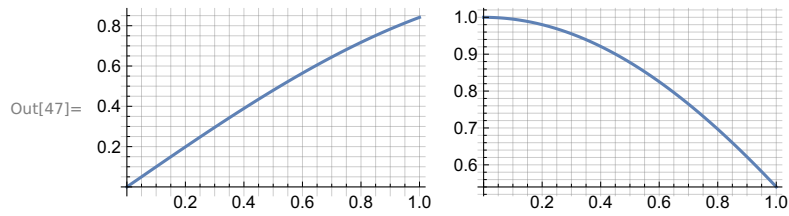
---

carryFirst

In[8]:= **? carryFirst**

carryFirst[func][args] return function f[♯♯♯, args] &

In[48]:= **carryFirst[f][y1, y2][x]**

Out[48]= f[x, y1, y2]

In[46]:= `plotter = carryFirst[Plot][{x, 0, 1}, GridLines → All];`

`{Sin[x], Cos[x]} // Map[plotter] // GraphicsRow`

Out[47]=

---

carryLast

In[9]:= `? carryLast`

carryLast[func][args] return function f[args, ▦] &

In[49]:= `carryLast[f][y1, y2][x]`

Out[49]= `f[y1, y2, x]`

In[34]:= `searcher = carryLast[Cases][{1, 2, 3, a, b, 4, 5}];`
`searcher[_Integer]`

Out[35]= `{1, 2, 3, 4, 5}`

In[36]:= `searcher[_Symbol]`

Out[36]= `{a, b}`

In[37]:= `searcher[1]`

Out[37]= `{1}`