PL.js Manual

Table of contents

e

```
pl.e(element);
```

element (string):

- id
    - use # in front of id
    - returns element object
- class
    - use . in front of classname
    - you can use index [n], [odd] or [even]
    - returns element object or array of element objects
- tagname
    - give just tagname
    - you can use index [n], [odd] or [even]
    - returns element object or array of element objects
- all elements
    - give *
    - returns element object

c

```
pl.c(element[, add, classname]);
```

element (string):
- like in `pl.e();`
  - only one element

add (boolean):
- if undefined
  - returns classlist of the element
- if true
  - adds classname to the classlist of the element
- if flase
  - removes classname from the classlist of the element

classname (string):
- if add is true or false, will be added or removed from the classlist of the element

val

```
pl.val(element[, newValue]);
```

element (string):

- like in `pl.e();`
    - only one element

newValue (string):

- will be set to elements innerHTML or value
    - value used with input and textarea
    - innerHTML with everything else

attr

```
pl.attr(element, attribute[, newValue]);
```

element (string):
- like in `pl.e();`
    - only one element

attribute (string):
- attribute that will be read or written

newValue (string):
- if undefined
    - returns current value of the attribute
- if defined
    - sets attribute to this value

css

```
pl.css(element, property[, newValue]);
```

element (string):
- like in `pl.e();`
  - only one element

property (string):
- CSS-property that will be read or written

newValue (string):
- if undefined:
  - returns current value of CSS-property
- if defined:
  - sets CSS-property to this value

ael

```
pl.ael(element, event, function);
```

element (string):

- like in `pl.e();`
  - only one element

event (string):

- event that triggers the function
  - without on-beginning (e.g. `onclick` is wrong but `click` is correct)

function (function):

- function that will be triggered

intvl

```
pl.intvl(identifier, interval, function, firstNow, 
rounds);
```

identifier (string):
- give an unique name for the interval

interval (integer):
- time between two rounds in milliseconds

function (string):
- name of the function that will be called every round

firstNow (boolean):
- if true:
  - first round will be executed right now
- if false:
  - first round will be executed after interval time

rounds (int):
- number of rounds to be done
  - if undefined, will be executed infinity

order

```
pl.order(array1, array2);
```

array1 (array, items integer):
- the array that makes the order for array2

array2 (array, items any):
- the array that will be ordered

returns the ordered array2

act

`pl.act(element, action);`

element (string):
- like in `pl.e();`
  - only one element
  - target of an action

action (string):
- name of the action
  - hide|show|toggle|invisible|visible|visToggle
    - hide, show and toggle use CSS-display
    - invisible, visible and visToggle use CSS-visibility

ani

`pl.ani(element, animation, duration);`

element (string):
- like in `pl.e();`
  - only one element

animation (string):
- name of the animation
  - `fade-out|fade-in`

duration (integer):
- duration of the animation

search

```
pl.search(items, keyword, method);
```

items (array, items string):
- items where will be searched

keyword (string):
- string that will be searched

method (integer):
- index of the searching method that will be used
  - 1; complete equivalence at anywhere in the item
  - 2; complete equivalence at the beginning of the item
  - 3; item and keyword exactly the same

returns an array of all matched items