# Core Features

## General Settings

All settings are controlled in the B.settings collection. For example, the default title for say messaging would be in the B.settigs.say.defTitle item. All setting values in the table below should be prefixed with B.settings.

|  |  |  |
| --- | --- | --- |
| Settings | Default | Description |
| .say.defTitle | ‘System Message’ | The title to be used in normal “say” message dialogs if no specific title is provided or that title is provided as either blank or null. |
| .ScrollingTable.  fieldBackgroundColor | ‘ivory’ | In a scrolling table, any area that does not have a row has this background color. |
| .ScrollingTable.  footerBackgroundColor | ‘gainsboro’ | Footer buttons defined for a scrolling table will have the background color defined here. |
| .ScrollingTable.  footerHoverColor | ‘aqua’ | Footer buttons defined for a scrolling table will have the hovered background color defined here. |
| .ScrollingTable.  JQTheme | True | Defines if scrolling tables should use the JQuery theme or use stylesheet properties to define general colors. |

## Page initialization

Any page that has the B library added will have the following things applied once all scripts have loaded:

1. Any DIV or FORM with the class BDialog applied will be turned into a JQuery dialog with the following properties:
   1. autoOpen = false: This setting causes the dialog to be created, but not opened at the time of creation. Use the openDialog method to open and close dialogs.
   2. resizable = false: User defined dialogs are not resizable by default. You can use standard JQuery code to change from this default if you need to.
   3. modal = true: This setting causes the dialog to “take over” the screen when it is opened. Note that the rest of the browser is not locked as a true Modal dialog would cause. This benefit is provided by JQuery dialog usage.
2. Any FORM with the class BDialog applied will have the default “submit” action prevented. This will allow the developer to hand any submit actions. You can still use the onsubmit event if you wish.
3. Any item with a “title” property will have that property converted to a JQuery tooltip in “tracking” mode (meaning that the tooltip moves with the mouse cursor).
4. All buttons (BUTTON, INPUT TYPE=’BUTTON”, INPUT TYPE=’SUBMIT’) will be rendered as a JQuery Button. Any button dynamically created in Javascript will also be rendered as a JQuery button.
5. Any INPUT TYPE=’TEXT’ or TEXTAREA element will have the spellcheck attribute turned off.
6. Finally, if there is a method named init()… it will be run. This is where you should place code that should be run at the time the page is ready to be interacted with.

## General Functions / Methods

|  |  |
| --- | --- |
| Function Name | Usage |
| B.months | An array of month names (‘January’, ‘February’…). It is used by date formatting functions described later, but may be used for other purposes by the developer. |
| B.days | An array of day names (‘Sunday’, ‘Monday’…). It is used by date formatting functions described later, but may be used for other purposes by the developer. |
| B.getDateParts(date) | When passed a valid date (or string that can be converted to a date using the Javascript new Date(date)), it returns a collection of values that can be used for formatting or other needs. The collection returned is: **M**: a number representing the month (1, 2, 3 .. 12) **MM**: a number for the month with leading zero to make 2 digits if necessary (01, 02, etc)  **MMM**: three character representation of month (‘Jan’, ‘Feb’…)  **MMMM**: full name of month (‘January’, ‘February’…) **D**: a number representing the day of month (1, 2 … 31)  **DD**: a number for the day of month with leading zero to make 2 digits if necessary (01, 02 … 31)  **DDD**: three character representation of day of week (‘Sun’,’Mon’…)  **DDDD**: full name of day of week (‘Sunday’,’Monday’…)  **YYYY**: The year using 4 digits  **H**: the hour of day starting – to be used along with the ap value (1 .. 12)  **HH**: two digit hour of the day (01 .. 12)  **ap**: Lower case a or p designating AM or PM time for the H and HH values  **NN**: two digit value for minutes  **SS**: two digit value for the seconds  **sss**: (lower case) milliseconds as a number from 0 to 999  The entire collection is returned {M:0, MM:’00’, MMM:’Jan’, MMMM:’January’, D:0, DD:’00’, DDD:’Mon’, DDDD:’Monday’, YYYY:2017, H:8, HH:’08’, ap:’p’, NN:18, SS:35, sss:273} |
| B.format | Collection of formatting methods defined below. When using these methods, call them as B.format.METHOD. For example, format a number to 3 decimal places, using B.format.DECIMALPLACES(num, 3). |
| .DECIMALPLACES(num, plcs) | Rounds any value to the number of places defined. If the num value is not numeric, it returns 0.000 (number of places). Examples: B.format.DECIMALPLACES(1,4) -> 1.0000 B.format.DECIMALPLACES(10/3, 2) -> 3.333  If the number passed in is < 1, a leading zero is applied. Example:  B.format.DECIMALPLACES(.5, 3) -> 0.500  This also adds commas in appropriate locations. Example:  B.format.DECIMALPLACES(15231, 2) -> 15,231.00 |
| .ASLINK(html) | Returns the html passed in surrounded in a <span class=’anchor> tag. The anchor css class is defined in B<ver>.css and displays as blue text that changes to green underlined when hovered. This can be used when the click behavior is defined at a higher level (such as in the B.Tree object but you want items to work like anchors. |
| .DOLLARS(num) | This is really just a shortcut to calling B.DECIMALPLACES(num,2) and putting a dollar sign at the beginning. Example:  B.format.DOLLARS(10) -> $10.00 |
| .COMMAS(num) | This is really just a shortcut to calling B.DECIMALPLACES(num,0) |
| .INT(num) | In general, this just uses parseInt(num), but handles cases where num is not numeric by just returning the value as it was passed. |
| .LEFTPAD(val,char,places) | Adds char to the left of val as many times till the string is at least places long. Examples:  B.format.LEFTPAD(25,’0’,5) -> ‘00025’  B.format.LEFTPAD(‘DAN’,’-‘,10) -> ‘-------DAN’  B.format.LEFTPAD(‘ROOSEVELT’,’\*’,5) -> ‘ROOSEVELT’ |
| .LEFTZEROPAD(val,places) | Shortcut to calling .LEFTPAD with ‘0’ as the character |
| .TRIM(txt) | Performs a full trim (leading and trailing) on the string |
| .UTRIM(txt) | Performs a B.format.TRIM() and then converts to upper case |
| .UPTRIM(txt) | Same as UTRIM – because lots of people try to use UP instead of just U |
|  |  |