

## 2 1. jQueryUI の活用

jQueryUI は、jQuery の便利な機能をまとめたツールです。UI についていますので、ユーザーインターフェースに特化したツールです。jQuery UI で提供されている機能を、大きく分類すると以下のようになります。

- (a) インタラクション…ドラッグ&ドロップ、リサイズを行います。
- (b) ウィジェット…アコーディオン、ダイアログ、タブの機能を提供します。
- (c) ヴィジュアル効果…アニメーション機能、エフェクト機能の強化をします。

## (1) インタラクション (ドラッグ操作) その1

## 【実施事項 01】

- ① 「12jQueryUI.zip」を解凍し、「wp2」フォルダに保存します。
- ② 「12jQuery01.html」をエディタで開きます。
- ③ ヒントを見ながら反転文字部分を埋めて、「12jQuery02.html」と名前を付けて保存しましょう。

L	filename:12jQuery01.html⇒12jQuery02.html	ヒント・補足
01	<!DOCTYPE html>	
02	<html>	
03	<head>	
04	<meta charset="UTF-8">	
05	<title>12jQuery02</title>	title 変更
06	<link href="jquery-ui.min.css" rel="stylesheet">	
07	<style>	jQuery-UI のスタイルシート
08	/*Q.01*/ {	/*Q.01*/
09	width: 150px;	id="div1" を対象に設定
10	height: 150px;	
11	background: red;	
12	}	
13	</style>	
14	<script src="jquery-3.6.1.min.js"></script>	jQuery ライブラリを利用
15	<script src="jquery-ui.min.js"></script>	jQuery-UI ライブラリを利用
16	<script>	
17	\$(function() {	
18	\$("#/*Q.01*/")/*Q.02*/();	/*Q.02*/
19	});	id="div1"オブジェクトに対して
20	</script>	.draggable()でドラッグが可能
21	</head>	
22	<body>	
23	<div id="div1">	
24	DIV だよー	
25	</div>	
26	</body>	
27	</html>	

- ③ 「12jQuery02.html」をブラウザで開き、表示される文字をドラッグアンドドロップしてみましょう。  
 .draggable()を利用して、対象の要素をドラッグアンドドロップすることができます。  
 もう一つ、確かめてみましょう。

## (2) インタラクション (ドラッグ操作) その2

## 【実施事項 02】

- ① 「12jQuery02.html」 をエディタで開きます。
- ② ヒントを見ながら**反転文字部分**を追記・更新して、「12jQuery03.html」と名前を付けて保存しましょう。  
(灰色文字部分は【実施事項 01】で更新した部分です。)

L	filename:12jQuery02.html⇒12jQuery03.html	ヒント・補足
01	<!DOCTYPE html>	
02	<html>	
03	<head>	
04	<meta charset="UTF-8">	
05	<title>12jQuery03</title>	title 変更
06	<link href="jquery-ui.min.css" rel="stylesheet">	
07	<style>	
08	div {	1 行更新 (対象を div にする)
09	width: 150px;	
10	height: 150px;	
11	float: left;	11～15 行目: 5 行追記
12	margin: 10px;	
13	color: #fff;	
14	text-align: center;	
15	line-height: 150px;	
16	background: red;	
17	}	
18	</style>	
19	<script src="jquery-3.6.1.min.js"></script>	
20	<script src="jquery-ui.min.js"></script>	
21	<script>	
22	\$(function() {	
23	\$("#div1").css({background:"blue"});	
24	\$("#div2").css({background:"green"});	
25	start : function (event , ui){	25～32 行目: 8 行追記
26	\$(this).css({background:"blue"});	id="div2" オブジェクトに対して
27	},	.draggable()でドラッグ (アンドド
28	stop : function (event , ui){	ロップ) を可能とし、
29	\$(this).css({background:"green"});	ドラッグがスタートしたら "blue"
30	}	ドラッグがストップしたら "green"
31	});	とする。
32	});	
33	</script>	
34	</head>	
35	<body>	39～41 行目: 3 行追記
36	<div id="div1">	処理対象として id="div2"を追加
37	DIV だよー	
38	</div>	
39	<div id="div2">	
40	これも DIV だよー	
41	</div>	
42	</body>	
43	</html>	

- ③ 「12jQuery03.html」 をブラウザで開き、表示される枠をドラッグアンドドロップしてみましょう。
- 対象のオブジェクトが動き始めたら: start の処理を行い、止まったら: stop の処理を行っていることがわかります。また、処理と処理の間は、(半角カンマ)で区切っています。

## (3) インタラクション (ドラッグ操作) その3

## 【実施事項 03】

- ① 「12jQuery03.html」 をエディタで開きます。
- ② ヒントを見ながら**反転文字部分**を追記・更新して、「12jQuery04.html」と名前を付けて保存しましょう。  
(灰色文字部分は【実施事項 01】で更新した部分です。)

L	filename:12jQuery03.html⇒12jQuery04.html	ヒント・補足
01	<!DOCTYPE html>	title 変更  スタイルシート部分は変更なしのため省略
02	<html>	
03	<head>	
04	<meta charset="UTF-8">	
05	<title>12jQuery04</title>	
06	<link href="jquery-ui.min.css" rel="stylesheet">	
07	<style>	
	(省略)	
18	</style>	
19	<script src="jquery-3.6.1.min.js"></script>	
20	<script src="jquery-ui.min.js"></script>	28～35 行目：8 行追記 ドラッグ中の処理を追加。 div オブジェクトの位置を表示。
21	<script>	
22	\$(function() {	
23	\$("/Q.01"/)/Q.02*/();	
24	\$("#div2")/Q.02*/ ({	
25	start : function (event , ui){	
26	\$(this).css({background:"blue"});	
27	},	
28	drag : function (event , ui){	
29	\$(this).text(	
30	"left:"	
31	+ \$(this).position().left	
32	+ ","	
33	+ "top:"	
34	+ \$(this).position().top);	
35	},	
36	stop : function (event , ui){	
37	\$(this).css({background:"green"});	
38	}	
39	});	
40	});	
41	</script>	
42	</head>	
43	<body>	
44	<div id="div1">	
45	DIV だよー	
46	</div>	
47	<div id="div2">	
48	これも DIV だよー	
49	</div>	
50	</body>	
51	</html>	

- ③ 「12jQuery04.html」 をブラウザで開き、表示される枠をドラッグアンドドロップしてみましょう。  
ドラッグしている最中は:drag の処理を行い、オブジェクトの位置を表示していることがわかります。

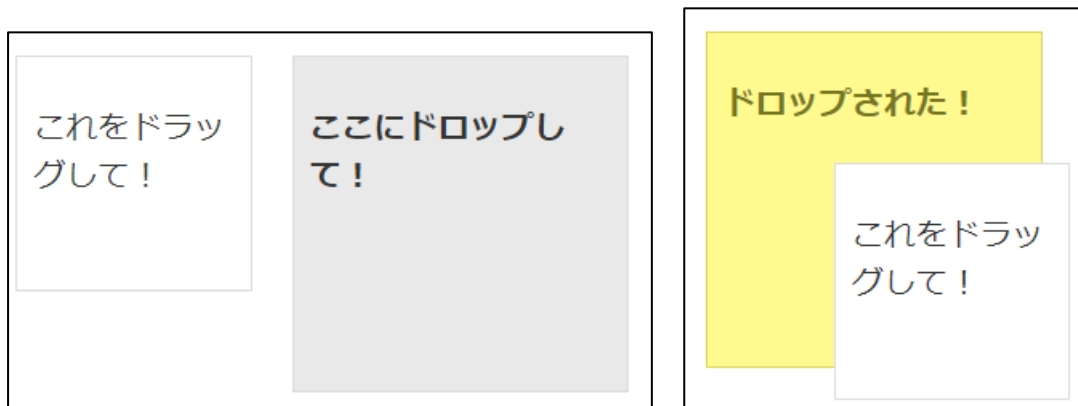
（4）インタラクション（その他）

その他以下のような処理を実現できる。

Draggable（ドラッグ後ドロップを検知して処理する）

【実施事項 04】

「12jQuery05.html」をブラウザで開き、「これをドラッグして！」と表示される枠を「ここにドロップして！」と表示される枠にドラッグアンドドロップしてみましょう。



Resizable（オブジェクトの大きさを変更する処理）

【実施事項 05】

- ① 「12jQuery06.html」をブラウザで開き、枠の右下をドラッグアンドドロップしてみましょう。
- ② 「12jQuery06.html」をエディタで開きましょう。22 行目の「.resizable」でサイズ変更を可能としています。  
23 行目 animate 属性により、枠を動かす処理にアニメーション効果がつきます。  
24 行目 helper 属性や css により、枠を動かす前に枠線（点線）を表示させています。  
25 行目 ghost 属性により、残像が表示されるようになっています。

Selectable（複数あるオブジェクトの中から選択する処理）

【実施事項 06】

- ① 「12jQuery07.html」をブラウザで開いてみましょう。  
選択されたものを上部に表示させるようにしています。
- ② 「12jQuery07.html」をエディタで開きましょう。33 行目の「.selectable」で選択可能としています。

Sortable（複数あるオブジェクトをドラッグアンドドロップで入れ替える処理）が実現できます。

【実施事項 07】

- ① 「12jQuery08.html」をブラウザで開いてみましょう。  
表示されたリストを入れ替えることができます。
- ② 「12jQuery08.html」をエディタで開きましょう。30 行目の「.sortable」で入れ替え可能としています。

## (5) ウィジェット (スライダー)

HTML5 で `<input type="range">` 要素を用いることで基本的なスライダーを作成できますが、jQueryUI の Slider ウィジェットにオプション設定を加えることで縦置きや範囲指定が可能となります。

## 【実施事項 08】

- ① 「12jQuery09.html」をエディタで開きます。
- ② ヒントを見ながら反転文字部分を追記・更新して、「12jQuery10.html」と名前を付けて保存しましょう。

L	filename:12jQuery09.html⇒12jQuery10.html	ヒント・補足
01	<code>&lt;!DOCTYPE html&gt;</code>	
02	<code>&lt;html&gt;</code>	
03	<code>&lt;head&gt;</code>	
04	<code>&lt;meta charset="UTF-8"&gt;</code>	
05	<code>&lt;title&gt;12jQuery10&lt;/title&gt;</code>	title 変更
06	<code>&lt;link href="jquery-ui.min.css" rel="stylesheet"&gt;</code>	
07	<code>&lt;style&gt;</code>	
08	<code>/*Q.03*/ {</code>	/*Q.03*/
09	<code>width: 300px;</code>	id="slider"を対象に設定
10	<code>}</code>	
11	<code>&lt;/style&gt;</code>	
12	<code>&lt;script src="jquery-3.6.1.min.js"&gt;&lt;/script&gt;</code>	jQuery ライブラリを利用
13	<code>&lt;script src="jquery-ui.min.js"&gt;&lt;/script&gt;</code>	jQuery-UI ライブラリを利用
14	<code>&lt;script&gt;</code>	
15	<code>\$(function() {</code>	
16	<code>\$(/*Q.03*/).slider({</code>	/*Q.03*/
17	<code>orientation: 'horizontal',</code>	id="slider"を対象に設定
18	<code>/*Q.04*/: 0,</code>	/*Q.04*/~/*Q.07*/
19	<code>/*Q.05*/: 100,</code>	最小値は min、最大値は max、
20	<code>/*Q.06*/: 2,</code>	増減数は step、初期値は
21	<code>/*Q.07*/: 50,</code>	value
22	<code>create: function(e, ui) {</code>	で指定することができる。
23	<code>\$(/*Q.08*/).val(\$(this).slider('option', 'value'));</code>	/*Q.08*/
24	<code>},</code>	スライダーの値が変更され
25	<code>change: function(e, ui) {</code>	たら、id="num"に値を設定
26	<code>\$(/*Q.08*/).val(ui.value);</code>	する。
27	<code>}</code>	
28	<code>});</code>	
29	<code>});</code>	
30	<code>&lt;/script&gt;</code>	
31	<code>&lt;/head&gt;</code>	
32	<code>&lt;body&gt;</code>	
33	<code>&lt;input id="num" type="text" size="3" readonly&gt;</code>	
34	<code>&lt;div id="slider"&gt;&lt;/div&gt;</code>	
35	<code>&lt;/body&gt;</code>	
36	<code>&lt;/html&gt;</code>	

- ③ 「12jQuery10.html」をブラウザで開きましょう。スライダーを左右に動かしてみましょう。

縦置きスライダーにするため、さらに手直ししてみましょう。

【実施事項 09】

- ① 「12jQuery10.html」 をエディタで開きます。
- ② ヒントを見ながら反転文字部分を追記・更新して、「12jQuery11.html」と名前を付けて保存しましょう。  
(灰色文字部分は【実施事項 08】で更新した部分です。)

L	filename:12jQuery10.html⇒12jQuery11.html	ヒント・補足
01	<!DOCTYPE html>	
02	<html>	
03	<head>	
04	<meta charset="UTF-8">	
05	<title>12jQuery11</title>	title 変更
06	<link href="jquery-ui.min.css" rel="stylesheet">	
07	<style>	
08	/*Q.03*/ {	
09	/*Q.09*/: 300px;	/*Q.09*/ 高さ指定する項目に変更する
10	}	
11	</style>	
12	<script src="jquery-3.6.1.min.js"></script>	
13	<script src="jquery-ui.min.js"></script>	
14	<script>	
15	\$(function() {	
16	\$("#/*Q.03*/").slider({	
17	orientation: /*Q.10*/,	/*Q.10*/ horizontal を vertical に変更する。
18	/*Q.04*/: 0,	
19	/*Q.05*/: 100,	
20	/*Q.06*/: 2,	
21	/*Q.07*/: 50,	
22	create: function(e, ui) {	
23	\$('#/*Q.08*/').val(\$(this).slider('option','value'));	
24	},	
25	change: function(e, ui) {	
26	\$('#/*Q.08*/').val(ui.value);	
27	}	
28	});	
29	});	
30	</script>	
31	</head>	
32	<body>	
33	<input id="num" type="text" size="3" readonly>	
34	<div id="slider"></div>	
35	</body>	
36	</html>	

- ③ 「12jQuery11.html」 をブラウザで開きましょう。スライダーを上下に動かしてみましょう。

さらに範囲スライダーにするため、さらに手直ししてみましょう。

### 【実施事項 10】

- ① 「12jQuery11.html」 をエディタで開きます。
- ② ヒントを見ながら**反転文字部分**を追記・更新して、「12jQuery12.html」と名前を付けて保存しましょう。  
(灰色文字部分は【実施事項 08】・【実施事項 09】で更新した部分です。)

L	filename:12jQuery11.html⇒12jQuery12.html	ヒント・補足
01	<!DOCTYPE html>	
02	<html>	
03	<head>	
04	<meta charset="UTF-8">	
05	<title>12jQuery12</title>	title 変更
06	<link href="jquery-ui.min.css" rel="stylesheet">	
07	<style>	
08	/*Q.03*/ {	
09	/*Q.09*/: 300px;	
10	}	
11	</style>	
12	<script src="jquery-3.6.1.min.js"></script>	
13	<script src="jquery-ui.min.js"></script>	
14	<script>	
15	\$(function() {	
16	\$("#Q.03").slider({	
17	orientation: /*Q.10*/,	
18	/*Q.04*/: 0,	
19	/*Q.05*/: 100,	
20	/*Q.06*/: 2,	
21	range: true,	range: true とすることで
22	values: [10,90],	範囲指定スライダーとなる。あ
23	create: function(e, ui) {	わせて初期値は value という変数で
24	var values=\$(this).slider('option', 'values')	はなく、values 配列で values:
25	\$("#min").val(values[0]);	[10,90] と配列の形で書く。
26	\$("#max").val(values[1]);	
27	},	24～26 行目は、スライダーの最小
28	change: function(e, ui) {	値・最大値の初期値を設定する。
29	\$("#min").val(ui.values[0]);	
30	\$("#max").val(ui.values[1]);	29～30 行目は、スライダーの値
31	}	が変更されたら最小値・最大値の
32	});	値を設定する。
33	});	
34	</script>	
35	</head>	
36	<body>	
37	最大値<input id="max" type="text" size="3" readonly>	37 行目は id="max" として最大値
38	<div id="slider"></div>	を表示する。
39	最小値<input id="min" type="text" size="3" readonly>	39 行目は id="min" として最小値
40	</body>	を表示する。
41	</html>	

- ③ 「12jQuery12.html」 をブラウザで開きましょう。スライダーを上下に動かしてみましょう。

スライダーを組み合わせると、以下のようなこともできます。


【実施事項 11】「12jQuery13.html」 をブラウザで開いてみましょう。

## (6) ウィジェット (Datepicker)

jQueryUI を利用すると、簡単に高機能なカレンダーUI を作成できます。jQuery の機能を説明する前に、HTML5 標準機能で実現可能な日付入力インターフェースを確かめます。

## 【実施事項 12】

- ① 「12jQuery14.html」 をエディタとブラウザで開きます。

L	filename:12jQuery14.html	結果イメージ
01	<!DOCTYPE html>	
02	<html>	
03	<head>	
04	<meta charset="UTF-8">	
05	<title>12jQuery14</title>	
06	</head>	
07	<body>	
08	<form action="" method="post">	
09	<input type="date" name="date">	
10	<input type="submit" value="送信">	
11	</form>	
12	</body>	
13	</html>	

HTML5 では input 要素 type="date" とすることで日付入力インターフェースが利用できます。

(さらに CSS を指定すると input 要素に日付選択の▲▼を付けたり、入力値リセットをしたりできます)

それでは、jQueryUI ではどうなるか確かめてみましょう。

## 【実施事項 13】

- ① 「12datepicker.zip」 を解凍し、解凍したファイルを「wp2」フォルダに保存します。  
 ② 「12jQuery15.html」 をエディタで開いて確かめましょう。

L	filename:12jQuery15.html	補足
01	<!DOCTYPE html>	jQuery jQueryUI jQueryUI 用 CSS datepicker-ja.js は、 DatePicker を日本向けに ローカライズしたファイル
02	<html>	
03	<head>	
04	<meta charset="UTF-8">	
05	<title>12jQuery15</title>	
06	<script src="jquery-3.6.1.min.js"></script>	
07	<script src="jquery-ui.min.js"></script>	
08	<link rel="stylesheet" href="jquery-ui.min.css">	
09	<script src="datepicker-ja.js"></script>	
10	<script>	
11	\$(function() {	
12	\$("#dp").datepicker();	
13	});	
14	</script>	
15	</head>	
16	<body>	
17	<p>Date:<input type="text" name="date" id="dp"></p>	
18	</body>	
19	</html>	

- ③ 「12jQuery15.html」 をブラウザで開いて確かめます。

右のように、英語表記になっているので修正しましょう。





さらにオプションなどを加え高機能なカレンダー入力を実現しましょう。

#### 【実施事項 14】

① 「12jQuery15.html」 をエディタで開きます。

② 反転文字部分を追記・更新して、「12jQuery16.html」 と名前を付けて保存しましょう。

L	filename:12jQuery15.html⇒12jQuery16.html	補足
01	<!DOCTYPE html>	<b>defaultDate</b>
02	<html>	デフォルトでハイライトされる日付
03	<head>	<b>dateFormat</b>
04	<meta charset="UTF-8">	入力される日付のフォーマット
05	<title>12jQuery15</title>	<b>showWeek</b>
06	<script src="jquery-3.6.1.min.js"></script>	左列にその年の何週目か表示
07	<script src="jquery-ui.min.js"></script>	<b>showOtherMonths</b>
08	<link rel="stylesheet" href="jquery-ui.min.css">	余剰部分があれば前後月の日を表示
09	<script src="datepicker-ja.js"></script>	表示
10	<script>	<b>selectOtherMonths</b>
11	\$(function() {	余剰部分があれば前後月の日を選択
12	\$("#dp").datepicker({	<b>changeYear</b>
13	defaultDate : new Date(),	年部分をプルダウン表示
14	dateFormat : 'yy/mm/dd(DD)',	<b>changeMonth</b>
15	showWeek : true,	月部分をプルダウン表示
16	showOtherMonths : true,	<b>numberOfMonths</b>
17	selectOtherMonths : true,	複数月を表示。[2,3] のように設定すると、2 行×3 行で月表示可
18	changeYear : true,	<b>showOn: "both"</b>
19	changeMonth : true,	テキストとボタンの両方で表示
20	numberOfMonths : [2,2],	<b>buttonText</b>
21	showOn : "both",	showOn で表示したボタンの名前
22	buttonText : "カレンダーを表示",	<b>showButtonPanel</b>
23	showButtonPanel : true,	今日ボタンと閉じるボタンを表示
24	currentText : "今日",	<b>currentText</b>
25	closeText : "閉じる",	今日ボタンの名前
26	});	<b>closeText</b>
27	\$("#dp").datepicker(	閉じるボタンの名前
28	"setDate", new Date());	<b>"setDate", new Date()</b>
29	});	オプション等と別に指定することでデフォルトに今日の日付を設定。
30	</script>	
31	</head>	
32	<body>	
33	<p>Date:<input type="text" name="date" id="dp"></p>	
34	</body>	
35	</html>	

③ 「12jQuery16.html」 をブラウザで開いて確かめましょう。

## 【出カイメージ】

Date: 2022/12/08(木曜日) カレンダーを表示

2022		年		12月				2023年 1月							2023年 2月												
週	日	月	火	水	木	金	土	週	日	月	火	水	木	金	土	週	日	月	火	水	木	金	土				
47	27	28	29	30	1	2	3	52	1	2	3	4	5	6	7	4	29	30	31	1	2	3	4				
48	4	5	6	7	8	9	10	1	8	9	10	11	12	13	14	5	5	6	7	8	9	10	11				
49	11	12	13	14	15	16	17	2	15	16	17	18	19	20	21	6	12	13	14	15	16	17	18				
50	18	19	20	21	22	23	24	3	22	23	24	25	26	27	28	7	19	20	21	22	23	24	25				
51	25	26	27	28	29	30	31	4	29	30	31	1	2	3	4	8	26	27	28	1	2	3	4				
2023年 3月							2023年 4月							2023年 5月													
週	日	月	火	水	木	金	土	週	日	月	火	水	木	金	土	週	日	月	火	水	木	金	土				
8	26	27	28	1	2	3	4	12	26	27	28	29	30	31	1	17	30	1	2	3	4	5	6				
9	5	6	7	8	9	10	11	13	2	3	4	5	6	7	8	18	7	8	9	10	11	12	13				
10	12	13	14	15	16	17	18	14	9	10	11	12	13	14	15	19	14	15	16	17	18	19	20				
11	19	20	21	22	23	24	25	15	16	17	18	19	20	21	22	20	21	22	23	24	25	26	27				
12	26	27	28	29	30	31	1	16	23	24	25	26	27	28	29	21	28	29	30	31	1	2	3				
								17	30	1	2	3	4	5	6	22	4	5	6	7	8	9	10				
<div>今日</div> <div>閉じる</div>																											

今日

閉じる

HTML5 の標準と比べて、かなりカスタマイズをすることができました。

## 【実施事項 15】⇒第 12 回目に提出する課題です。

土日の色を付ける処理を加えましょう。処理の流れは、以下の通りです。

- ・土日・平日表示用のクラス名に対するスタイルシートを用意します。
- ・取得した日付の曜日が 0 ならば日曜のクラス名を、6 ならば土曜のクラス名を、それ以外であれば平日のクラス名を割り当てます。

① 「12jQuery16.html」 をエディタで開きます。

② 次ページの反転文字部分を追記・更新し、ブラウザで表示を確認しましょう。

(灰色文字部分は【実施事項 14】で更新した部分です。)

③ &lt;title&gt;12jQuery by1TX YY&lt;/title&gt;の、

- ・1TX には、自分のクラスを半角文字で
- ・YY には、出席番号を半角文字で

更新してください。

④ 記述が終わったら、12jQuery.html と名前を付けて保存します。

⑤ この「12jQuery.html」を提出してください。

L	filename:12jQuery16.html⇒12jQuery.html	補足
01	<!DOCTYPE html>	
02	<html>	
03	<head>	
04	<meta charset="UTF-8">	
05	<title>12jQuery by1TX YY</title>	
06	<script src="jquery-3.6.1.min.js"></script>	
07	<script src="jquery-ui.min.js"></script>	
08	<link rel="stylesheet" href="jquery-ui.min.css">	
09	<style>	
10	.ui-datepicker-calendar .day-sunday > a {	日曜用のスタイルシート
11	background: #ffc0c0;	
12	}	
13	.ui-datepicker-calendar .day-saturday > a {	土曜用のスタイルシート
14	background: #c0c0ff;	
15	}	
16	.ui-datepicker-calendar .day-weekday > a {	平日用のスタイルシート
17	background: #c0ffc0;	
18	}	
19	</style>	
20	<script src="datepicker-ja.js"></script>	
21	<script>	
22	\$(function() {	
23	\$("#dp").datepicker({	
24	defaultDate : new Date(),	
25	dateFormat : 'yy/mm/dd(DD)',	
26	showWeek : true,	
27	showOtherMonths : true,	
28	selectOtherMonths : true,	
29	changeYear : true,	
30	changeMonth : true,	
31	numberOfMonths : [2,2],	
32	showOn : "both",	
33	buttonText : "カレンダーを表示",	
34	showButtonPanel : true,	
35	currentText : "今日",	
36	closeText : "閉じる",	
37	beforeShowDay: function(date) {	取得した曜日によって、
38	if (date.getDay() == 0) {	beforeShowDay オプション
39	return [true, 'day-sunday', null];	の引数を return にて割り
40	} else if (date.getDay() == 6) {	当てています。
41	return [true, 'day-saturday', null];	引数 1 : true ならば選択可
42	}	引数 2 : class 属性の設定
43	return [true, 'day-weekday', null];	引数 3 : その日を表すツール
44	}	チップ(マウスオーバーすると
45	});	表示されるような情報)
46	\$("#dp").datepicker("setDate", new Date());	
47	});	
48	</script>	
49	</head>	
50	<body>	
51	<p>Date:<input type="text" name="date" id="dp"></p>	
52	</body>	
53	</html>	

## 【出カイメージ】

Date: 2022/12/08(木曜日) カレンダーを表示

2022		年		12月				2023年 1月							2023年 2月									
週	日	月	火	水	木	金	土	週	日	月	火	水	木	金	土	週	日	月	火	水	木	金	土	
47	27	28	29	30	1	2	3	52	1	2	3	4	5	6	7	4	29	30	31	1	2	3	4	
48	4	5	6	7	8	9	10	1	8	9	10	11	12	13	14	5	5	6	7	8	9	10	11	
49	11	12	13	14	15	16	17	2	15	16	17	18	19	20	21	6	12	13	14	15	16	17	18	
50	18	19	20	21	22	23	24	3	22	23	24	25	26	27	28	7	19	20	21	22	23	24	25	
51	25	26	27	28	29	30	31	4	29	30	31	1	2	3	4	8	26	27	28	1	2	3	4	
2023年 3月							2023年 4月							2023年 5月										
週	日	月	火	水	木	金	土	週	日	月	火	水	木	金	土	週	日	月	火	水	木	金	土	
8	26	27	28	1	2	3	4	12	26	27	28	29	30	31	1	17	30	1	2	3	4	5	6	
9	5	6	7	8	9	10	11	13	2	3	4	5	6	7	8	18	7	8	9	10	11	12	13	
10	12	13	14	15	16	17	18	14	9	10	11	12	13	14	15	19	14	15	16	17	18	19	20	
11	19	20	21	22	23	24	25	15	16	17	18	19	20	21	22	20	21	22	23	24	25	26	27	
12	26	27	28	29	30	31	1	16	23	24	25	26	27	28	29	21	28	29	30	31	1	2	3	
								17	30	1	2	3	4	5	6	22	4	5	6	7	8	9	10	
<div>今日</div>																							<div>閉じる</div>	

土日・平日に色付けすることができました。

(参考) Holidays JP API を用いた祝日部分の色付け

Holidays JP API から JSON 形式で祝日データを取得し、祝日であれば祝日用のクラス名を割り当てる、といった処理を加えれば実現できます。

Holidays JP API <https://holidays-jp.github.io/>

L	filename:12jQuery.html⇒12jQuery18.html (抜粋)	補足
10 11 12 13	<pre> .ui-datepicker-calendar .day-sunday &gt; a, .ui-datepicker-calendar .day-holiday &gt; a {     background: #ffc0c0; } </pre>	日曜と祝日は同じスタイルシートとしている。
24  45 46 47 48 49 50 51 52 53	<pre> \$.get("https://holidays-jp.github.io/api/v1/date.json", function(holidaysData) {      var holidays = Object.keys(holidaysData);     for (var i = 0; i &lt; holidays.length; i++) {         var holiday = new Date(Date.parse(holidays[i]));         if (holiday.getYear() == date.getYear() &amp;&amp;             holiday.getMonth() == date.getMonth() &amp;&amp;             holiday.getDate() == date.getDate()) {             return [true, 'day-holiday', null];         }     } } </pre>	Holidays JP API から JSON データ取得している。 祝日データに合致する場合に、祝日用クラスを割り当てている。
58	});	24 行との対。

### 【出力イメージ】

Date:

2022 年 12 月							2023 年 1 月							2023 年 2 月									
週	日	月	火	水	木	金	土	週	日	月	火	水	木	金	土	週	日	月	火	水	木	金	土
47	27	28	29	30	1	2	3	52	1	2	3	4	5	6	7	4	29	30	31	1	2	3	4
48	4	5	6	7	8	9	10	1	8	9	10	11	12	13	14	5	5	6	7	8	9	10	11
49	11	12	13	14	15	16	17	2	15	16	17	18	19	20	21	6	12	13	14	15	16	17	18
50	18	19	20	21	22	23	24	3	22	23	24	25	26	27	28	7	19	20	21	22	23	24	25
51	25	26	27	28	29	30	31	4	29	30	31	1	2	3	4	8	26	27	28	1	2	3	4

2023 年 3 月							2023 年 4 月							2023 年 5 月									
週	日	月	火	水	木	金	土	週	日	月	火	水	木	金	土	週	日	月	火	水	木	金	土
8	26	27	28	1	2	3	4	12	26	27	28	29	30	31	1	17	30	1	2	3	4	5	6
9	5	6	7	8	9	10	11	13	2	3	4	5	6	7	8	18	7	8	9	10	11	12	13
10	12	13	14	15	16	17	18	14	9	10	11	12	13	14	15	19	14	15	16	17	18	19	20
11	19	20	21	22	23	24	25	15	16	17	18	19	20	21	22	20	21	22	23	24	25	26	27
12	26	27	28	29	30	31	1	16	23	24	25	26	27	28	29	21	28	29	30	31	1	2	3
								17	30	1	2	3	4	5	6	22	4	5	6	7	8	9	10

今日