

Imperial.StyleForm

Opis

Plugin pozwala zastąpić natywne elementy formularza takie jak przełączniki (checkboxy), opcje wyboru (radio), listy rozwijalne (select), przyciski dodawania nowego pliku (file), kodem html umożliwiającym w prosty sposób przygotowanie ich wyglądu graficznego poprzez nadanie odpowiednich stylów.

Opis autora.

Dla walidacji formularza za pomocą pluginu jquery validation dodano odpowiednie wsparcie dzięki czemu klasa błędu pojawia się również na utworzonym elemencie.

Jak zacząć?

Z pluginu możemy skorzystać na dwa sposoby:

1. Dla formularza lub elementu który zawiera w sobie natywne elementy formularza możemy uruchomić metodę styleForm.

```
$(function(){  
    $("form").styleForm()  
});
```

2. Dla każdego z elementów formularza możemy uruchomić metodę oddzielnie z różnymi parametrami, służą do tego funkcję:

- a. Checkbox - checkboxStyle,

```
$(function(){  
    $("input[type='checkbox']").checkboxStyle()  
});
```

- b. Radio - radioStyle

```
$(function(){  
    $("input[type='radio']").radioStyle()  
});
```

- c. Select - selectStyle

```
$(function(){  
    $("select").selectStyle()  
});
```

d. File - fileUploaderStyle

```
$(function(){  
    $("input[type='file']").fileUploaderStyle()  
});
```

Możliwości funkcji checkboxStyle

1. Funkcja checkboxStyle może przyjmować następujące parametry:

- hover** - przyjmuje wartości typu boolean (domyślnie false), jeżeli opcja jest włączona to dodawana jest klasa i-ch-hover po najechaniu na ten element, i usuwana po opuszczeniu myszki z tego elementu,
- disabled** - przyjmuje wartości typu boolean (domyślnie false), jeżeli opcja jest włączona to wszystkim wybranym elementom zostanie przypisana wartość disabled. Elementy które są disabled mają dodawaną klasę i-ch-disabled,
- wrapClass** - może przyjmować wartości typu boolean, string, oraz wartość null. Jeżeli wprowadzimy wartość typu boolean i wybierzemy wartość true to wszystkie klasy z atrybutu class, które były ustawione na elemencie typu checkbox zostaną dodane do elementu i-ch-wrapper. Jeżeli wprowadzimy wartość jako string to do atrybutu class elementu i-ch-wrapper zostaną dodane wartości jakie podamy. Wartość null oznacza że nic nie zostanie dodane do atrybutu class dla elementu i-ch-wrapper.
- inputClass** - przyjmuje wartości typu string lub wartość null. Jeżeli wprowadzimy wartość jako string to do atrybutu class elementu checkbox zostaną dodane wartości jakie podamy.
- wrapTitle** - może przyjmować wartości typu boolean, string, oraz wartość null. Jeżeli wprowadzimy wartość typu boolean i wybierzemy wartość true to Tytuł z atrybutu title, który był ustawiony na elemencie typu checkbox zostanie ustawiony dla elementu i-ch-wrapper. Jeżeli wprowadzimy wartość jako string to dla atrybutu title elementu i-ch-wrapper zostanie ustawiona wartość jaka podamy. Wartość null oznacza że nie zostanie ustawiony atrybut title dla elementu i-ch-wrapper.
- ignoredClass** - przyjmuje wartości typu string. Podajmy tutaj klasy dla których elementy nie będą używać pluginu do formatowania.
- onEnter** - funkcja jaką można wykonać podczas najechania na element przez myszkę,
- onLeave** - funkcja jaką można wykonać podczas opuszczenia elementu przez myszkę,
- onChange** - funkcja jaką można wykonać podczas zmiany wartości dla elementu
- onCreate** - funkcja jaką można wykonać podczas tworzenia elementu
- onClick** - funkcja jaką można wykonać po kliknięciu w element.

2. Najważniejsze klasy funkcji checkboxStyle:

- i-ch-wrapper** - kontener dla wszystkich elementów dotyczących checkboxa.
- i-ch-disabled** - tą klasę przyjmuje dodatkowo kontener w przypadku kiedy checkbox posiada atrybut disabled,
- i-ch-default** - klasa ta jest elementem który odpowiada od strony graficznej za wygląd naszego checkbox,
- i-ch-clicked** - klasa ta jest dodawana w momencie kiedy checkbox jest w stanie aktywnym jest ona dodawana do elementu i-ch-default,
- i-ch-hover** - klasa dodawana jest do konteneru i-ch-wrapper, na jej podstawie możemy określić wygląd naszego elementu po najechaniu myszką na niego,
- i-ch-only-checkbox** - ta klasa występuje w przypadku kiedy checkbox nie posiada żadnej etykiety można za jej pomocą usunąć nie potrzebny margines.

3. Dodatkowe funkcje które mogą być wywołane po inicjalizacji:

- destroy** - przywraca checkbox do stanu pierwotnego, przed zastosowaniem pluginu,
- refresh** - przywraca checkbox do stanu pierwotnego, przed zastosowaniem pluginu i tworzy go na nowo.

4. Przykłady wywołania funkcji:

```
$(function() {  
    $("input[type='checkbox']").checkboxStyle(  
    {  
        hover: true,  
        disabled: false,  
        wrapClass: "additionalWrapClassName",  
        inputClass: "additionalInputClassName",  
        wrapTitle: true,  
        ignoredClass: "ignoreCheckbox",  
        onEnter: function()  
        {  
            /*kod do wykonania przy najechaniu myszą*/  
        },  
        onLeave: function()  
        {  
            /*kod do wykonania po opuszczeniu myszy z elementu*/  
        },  
        onChange: function()  
        {  
            /*kod do wykonania w momencie zmiany wartości*/  
        },  
        onCreate: function()  
        {  
            /*kod do wykonania w momencie tworzenia kodu html elementu*/  
        },  
        onClick: function()  
        {  
            /*kod do wykonania w momencie kliknięcia na element*/  
        }  
    });  
  
$(function(){  
    $("input[type='checkbox']").checkboxStyle("destroy");  
});  
  
$(function(){  
    $("input[type='checkbox']").checkboxStyle("refresh");  
});
```

Możliwości funkcji radioStyle

1. Funkcja radioStyle może przyjmować następujące argumenty:

- hover** - przyjmuje wartości typu boolean (domyślnie false), jeżeli opcja jest włączona to dodawana jest klasa i-r-hover po najechaniu na ten element, i usuwana po opuszczeniu myszki z tego elementu,
- disabled** - przyjmuje wartości typu boolean (domyślnie false), jeżeli opcja jest włączona to wszystkim wybranym elementom zostanie przypisana wartość disabled. Elementy które są disabled mają dodawaną klasę i-r-disabled,
- wrapClass** - może przyjmować wartości typu boolean, string, oraz wartość null. Jeżeli wprowadzimy wartość typu boolean i wybierzemy wartość true to wszystkie klasy z atrybutu class, które były ustawione na elemencie typu radio zostaną dodane do elementu i-r-wrapper. Jeżeli wprowadzimy wartość jako string to do atrybutu class elementu i-r-wrapper zostaną dodane wartości jakie podamy. Wartość null oznacza że nic nie zostanie dodane do atrybutu class dla elementu i-r-wrapper.
- inputClass** - przyjmuje wartości typu string lub wartość null. Jeżeli wprowadzimy wartość jako string to do atrybutu class elementu radio zostaną dodane wartości jakie podamy.
- wrapTitle** - może przyjmować wartości typu boolean, string, oraz wartość null. Jeżeli wprowadzimy wartość typu

boolean i wybierzemy wartość true to Tytuł z atrybutu title, który był ustawiony na elemencie typu radio zostanie ustawiony dla elementu i-r-wrapper. Jeżeli wprowadzimy wartość jako string to dla atrybutu title elementu i-r-wrapper zostanie ustawiona wartość jaka podamy. Wartość null oznacza że nie zostanie ustawiony atrybut title dla elementu i-r-wrapper.

- f. **ignoredClass** - przyjmuje wartości typu string. Podajmy tutaj klasy dla których elementy nie będą używać pluginu do formatowania.
- g. **onEnter** - funkcja jaką można wykonać podczas najechania na element przez myszkę,
- h. **onLeave** - funkcja jaką można wykonać podczas opuszczenia elementu przez myszkę,
- i. **onChange** - funkcja jaką można wykonać podczas zmiany wartości dla elementu
- j. **onCreate** - funkcja jaką można wykonać podczas tworzenia elementu
- k. **onClick** - funkcja jaką można wykonać po kliknięciu w element.

2. Najważniejsze klasy funkcji checkboxStyle:

- a. **i-r-wrapper** - kontener dla wszystkich elementów dotyczących radio.
- b. **i-r-disabled** - tą klasę przyjmuje dodatkowo kontener w przypadku kiedy radio posiada atrybut disabled,
- c. **i-r-default** - klasa ta jest elementem który odpowiada od strony graficznej za wygląd naszego radio,
- d. **i-r-clicked** - klasa ta jest dodawana w momencie kiedy radio jest w stanie aktywnym jest ona dodawana do elementu i-ch-default,
- e. **i-r-hover** - klasa dodawana jest do konteneru i-ch-wrapper, na jej podstawie możemy określić wygląd naszego elementu po najechaniu myszką na niego.
- f. **i-r-only-radio** - ta klasa występuje w przypadku kiedy radio nie posiada żadnej etykiety można za jej pomocą usunąć nie potrzebny margines.

3. Dodatkowe funkcje które mogą być wywołane po inicjalizacji:

- a. **destroy** - przywraca radio do stanu pierwotnego, przed zastosowaniem pluginu.
- b. **refresh** - przywraca radio do stanu pierwotnego, przed zastosowaniem pluginu i tworzy go na nowo.

4. Przykłady wywołania funkcji:

```
$(function () {
    $("input[type='radio']").radioStyle(
    {
        hover: true,
        disabled: false,
        wrapClass: "additionalWrapClassName",
        inputClass: "additionalInputClassName",
        wrapTitle: true,
        ignoredClass: "ignoreRadio",
        onEnter: function ()
        {
            /*kod do wykonania przy najechaniu myszą*/
        },
        onLeave: function ()
        {
            /*kod do wykonania po opuszczeniu myszy z elementu*/
        },
        onChange: function ()
        {
            /*kod do wykonania w momencie zmiany wartości*/
        },
        onCreate: function ()
        {
            /*kod do wykonania w momencie tworzenia kodu html elementu*/
        },
        onClick: function ()
        {
            /*kod do wykonania w momencie kliknięcia na element*/
        }
    });
});

$(function() {
    $("input[type='checkbox']").radioStyle("destroy");
});

$(function() {
    $("input[type='checkbox']").radioStyle("refresh");
});
```

Możliwości funkcji selectStyle

1. Funkcja selectStyle może przyjmować następujące argumenty:

- hover** - przyjmuje wartości typu boolean (domyślnie false), jeżeli opcja jest włączona to dodawana jest klasa i-s-hover po najechaniu na ten element, i usuwana po opuszczeniu myszki z tego elementu,
- isClickable** - przyjmuje wartości typu boolean (domyślnie true), jeżeli ustawiony na true to oznacza że cały element listy jest klikalny żeby ją rozwinąć, jeżeli jest false to tylko handler i-s-button jest odpowiedzialny za rozwijanie listy.
- disabled** - przyjmuje wartości typu boolean (domyślnie false), jeżeli opcja jest włączona to wszystkim wybranym elementom zostanie przypisana wartość disabled. Elementy które są disabled mają dodawaną klasę i-s-disabled,
- wrapClass** - może przyjmować wartości typu boolean, string, oraz wartość null. Jeżeli wprowadzimy wartość typu boolean i wybierzemy wartość true to wszystkie klasy z atrybutu class, które były ustawione na elemencie typu select zostaną dodane do elementu i-s-wrapper. Jeżeli wprowadzimy wartość jako string to do atrybutu class elementu i-s-wrapper zostaną dodane wartości jakie podamy. Wartość null oznacza że nic nie zostanie dodane do atrybutu class dla elementu i-s-wrapper.
- inputClass** - przyjmuje wartości typu string lub wartość null. Jeżeli wprowadzimy wartość jako string to do atrybutu class elementu select zostaną dodane wartości jakie podamy.
- wrapTitle** - może przyjmować wartości typu boolean, string, oraz wartość null. Jeżeli wprowadzimy wartość typu boolean i wybierzemy wartość true to Tytuł z atrybutu title, który był ustawiony na elemencie typu select zostanie

ustawiony dla elementu i-s-wrapper. Jeżeli wprowadzimy wartość jako string to dla atrybutu title elementu i-s-wrapper zostanie ustawiona wartość jaka podamy. Wartość null oznacza że nie zostanie ustawiony atrybut title dla elementu i-s-wrapper.

- g. **ignoredClass** - przyjmuje wartości typu string. Podajmy tutaj klasy dla których elementy nie będą używać pluginu do formatowania.
- h. **minWidth** - przyjmuje wartości typu boolean (domyślnie true), jeżeli jest ustawiona wartość true to plugin wykorzystuje algorytm do określenie minimalnej szerokości elementu na podstawie wszystkich elementów znajdujących się na liście.
- i. **onEnter** - funkcja jaką można wykonać podczas najechania na element przez myszkę,
- j. **onLeave** - funkcja jaką można wykonać podczas opuszczenia elementu przez myszkę,
- k. **onChange** - funkcja jaką można wykonać podczas zmiany wartości dla elementu
- l. **onCreate** - funkcja jaką można wykonać podczas tworzenia elementu
- m. **onClick** - funkcja jaką można wykonać po kliknięciu w element.

2. Najważniejsze klasy funkcji checkboxStyle:

- a. **i-s-wrapper** - kontener dla wszystkich elementów dotyczących selecta.
- b. **i-s-disabled** - tą klasę przyjmuje dodatkowo kontener w przypadku kiedy select posiada atrybut disabled,
- c. **i-s-holder** - kontener dla elementu etykiety wybranej opcji oraz przycisku do rozwijania listy,
- d. **i-s-button** - przycisk do rozwijania / zwiijania listy
- e. **i-s-list** - kontener listy elementów naszej listy z select, mogą tutaj być elementy option oraz optgroup,
- f. **i-s-optgroup-list** - kontener listy elementów dla tagu optgroup
- g. **i-s-optgroup-label** - etykieta dla grupy elementów pochodząca z atrybutu label z tagu optgroup
- h. **i-s-default** - klasa określająca elementy listy możliwe do wybrania,
- i. **i-s-active** - klasa określa który element jest wybrany na liście,
- j. **i-s-optgroup** - kontener obejmujący grupę elementów listy oraz etykietę grupy jeżeli podano - tag optgroup.

3. Dodatkowe funkcje które mogą być wywołane po inicjalizacji:

- a. **destroy** - przywraca select do stanu pierwotnego, przed zastosowaniem pluginu.
- b. **refresh** - przywraca select do stanu pierwotnego, przed zastosowaniem pluginu i tworzy go na nowo.

4. Przykładowe wywołania funkcji:

```
$(function() {  
    $("select").selectStyle(  
    {  
        hover: true,  
        isClickable: true,  
        disabled: false,  
        wrapClass: "additionalWrapClassName",  
        inputClass: "additionalInputClassName",  
        wrapTitle: true,  
        ignoredClass: "ignoreSelect",  
        minWidth : true,  
        onEnter: function()  
        {  
            /*kod do wykonania przy najechaniu myszą*/  
        },  
        onLeave: function()  
        {  
            /*kod do wykonania po opuszczeniu myszy z elementu*/  
        },  
        onChange: function()  
        {  
            /*kod do wykonania w momencie zmiany wartości*/  
        },  
        onCreate: function()  
        {  
            /*kod do wykonania w momencie tworzenia kodu html elementu*/  
        }  
    })  
})
```

```
},
onClick: function()
{
    /*kod do wykonania w momencie kliknięcia na element*/
}
});

$(function(){
    $("select").selectStyle("destroy");
});

$(function(){
    $("select").selectStyle("refresh");
});
```

Możliwości funkcji fileUploaderStyle

1. Funkcja fileUploaderStyle może przyjmować następujące argumenty:
 - a. **class** - obiekt który może zawierać dwa elementy,
 - b. **class.wrap** - możemy podać dodatkową klasę dla kontenera i-file-upload-btn-wrap,
 - c. **class.btn** - możemy dodawać dodatkową klasę dla przycisku i-file-upload-btn-handler
2. Najważniejsze klasy funkcji fileUploaderStyle:
 - a. **i-file-upload-btn-wrap** - kontener dla przycisku typu file,
 - b. **i-file-upload-btn-handler** - przycisk pokazujący okno dodawania pliku.
3. Dodatkowe funkcje które mogą być wywołane po inicjalizacji:
 - a. **destroy** - przywraca przyciski do stanu pierwotnego, przed zastosowaniem pluginu.
4. Przykład wywołania funkcji:

```
$(function() {
    $("input[type='file']").fileUploaderStyle(
    {
        class: {
            wrap: "additionalClassWrap",
            btn: "additionalClassBtn"
        }
    });
});

$(function(){
    $("input[type='file']").selectStyle("destroy");
});
```

Możliwości funkcji styleForm

1. Funkcja styleForm może przyjmować następujące argumenty:
 - a. **checkbox** - obiekt który może zawierać wszystkie parametry jakie posiada funkcja checkboxStyle, zawiera dodatkowy parametr o nazwie **use** który określa czy użyć funkcji checkboxStyle dla wszystkich elementów checkbox które są dostępne w ramach obiektu na którym funkcja została wywołana,

- b. **radio** - obiekt który może zawierać wszystkie parametry jakie posiada funkcja `radioStyle`, zawiera dodatkowy parametr o nazwie **use** który określa czy użyć funkcji `checkboxStyle` dla wszystkich elementów checkbox które są dostępne w ramach obiektu na którym funkcja została wywołana,
- c. **select** - obiekt który może zawierać wszystkie parametry jakie posiada funkcja `selectStyle`, zawiera dodatkowy parametr o nazwie **use** który określa czy użyć funkcji `checkboxStyle` dla wszystkich elementów checkbox które są dostępne w ramach obiektu na którym funkcja została wywołana,,
- d. **fileUploader**- obiekt który może zawierać wszystkie parametry jakie posiada funkcja `fileUploaderStyle`, zawiera dodatkowy parametr o nazwie **use** który określa czy użyć funkcji `checkboxStyle` dla wszystkich elementów checkbox które są dostępne w ramach obiektu na którym funkcja została wywołana,

2. Dodatkowe funkcje które mogą być wywołane po inicjalizacji:

- a. **initValidationAdditionalEvent** - pozwala uruchomić event który będzie sprawdzał czy dany element typu radio, checkbox, select jest błędnie zwalidowany i przekazanie tej klasy walidatora dla błędnych pól do kontenera elementu np. `i-ch-wrapper`,
- b. **destroy** - resetuje formularz do stanu przed zastosowaniem pluginu.

3. Przykład wywołania funkcji:

```
$(function() {  
  $("form").styleForm(  
    {  
      checkbox: {  
        use: true,  
        /*Pozostałe parametry z funkcji checkboxStyle*/  
      },  
      radio: {  
        use: true,  
        /*Pozostałe parametry z funkcji radioStyle*/  
      },  
      select: {  
        use: true,  
        /*Pozostałe parametry z funkcji selectStyle*/  
      },  
      fileUploader: {  
        use: true,  
        /*Pozostałe parametry z funkcji fileUploaderStyle*/  
      }  
    }  
  );  
});
```

```
$(function(){  
  $("form").styleForm("initValidationAdditionalEvent");  
});
```

```
$(function(){  
  $("form").styleForm("destroy");  
});
```

maxlistelement

W ramach serwisu stosujemy pliki cookies by umożliwić Ci wygodne korzystanie z serwisu. Jeśli nie zmienisz ustawień dotyczących cookies w Twojej przeglądarce, będą one umieszczane na Twoim komputerze. W każdej chwili możesz zmienić swoje ustawienia. Dowiedz się więcej w naszej Polityce Prywatności

[Akceptuję](#)