

# Technologie internetowe w programowaniu

HTML5 Formularze, czyli co na to użytkownik

# Co nadal jest takie samo?

- ▶ Elementy formularza nadal powinny być zawarte w znaczniku `<form>`.
- ▶ Formularze w dalszym ciągu przesyłają wartości elementów do serwera gdy użytkownik wysyła formularz.
- ▶ Wszystkie znane dotychczas komponenty – pola tekstowe, pola jednokrotnego i wielokrotnego wyboru, guzki – nadal są dostępne i działają tak jak dotychczas (choć wzbogacone o nowe możliwości).
- ▶ Wszystkie elementy formularza można dowolnie obsługiwać i zmieniać z poziomu języka skryptowego.

# Nowe typy znacznika input

Typ	Reprezentacja
<code>tel</code>	Numer telefonu
<code>email</code>	Adres e-mail
<code>url</code>	Lokalizacja w sieci Web
<code>search</code>	Pole do wyszukiwania
<code>range</code>	Pasek horyzontalny do wyboru wartości numerycznych
<code>number</code>	Pole zawierające wyłącznie wartość numeryczną
<code>color</code>	Wybór koloru
<code>datetime</code>	Pełna data i czas + strefa czasowa
<code>datetime-local</code>	Pełna data i czas bez uwzględnienia strefy czasowej
<code>time</code>	Czas wraz z możliwością wyboru
<code>date</code>	Selektor konkretnej daty w kalendarzu
<code>week</code>	Selektor konkretnego tygodnia danego roku
<code>month</code>	Selektor konkretnego miesiąca danego roku

# Nowe atrybuty

## ▶ Placeholder

```
<label>Dane:</label>
```

```
<input name="dane" placeholder="Wpisz imię i nazwisko"><br/>
```

## ▶ Autocomplete

```
<label>PESEL:</label>
```

```
<input type="text" name="pesel" autocomplete="off">
```

## ▶ Autofocus

```
<label>PESEL:</label>
```

```
<input type="text" autofocus>
```

## ▶ Spellcheck

```
<textarea spellcheck="false">
```

## ▶ Min, Max, Step

```
<input id="zarobki" type="range" min="0" max="1000" step="200" value="200">
```

## ▶ Required

```
<input type="text" id="nazwisko" required>
```

## ▶ List

```
<datalist id="listaKontaktow">
```

```
  <option value="cos@cos.pl" label="Jan Nowak">
```

```
  <option value="gmail@norris.com" label="Chuck Norris">
```

```
</datalist>
```

```
<input type="email" id="kontakty" list="listaKontaktow"><br/>
```

# Walidacja formularza

## ▶ Status elementu formularza:

```
var sprawdz = document.idForm.idInput.validity;  
sprawdz.valid;
```

## ▶ Statusy walidacji:

- **valueMissing** - zwraca true jeśli brak wartości dla pola z atrybutem **required**.
- **typeMismatch** - sprawdza czy pola typu number, email, itp. mają dane w poprawnym formacie.
- **patternMismatch** - sprawdza czy pole ma dane w zdefiniowanym wcześniej formacie (przez **wyrażenie regularne**).

```
<input type= "text" name= "kartakred" pattern="[0-9]{16}"  
title="Musi być 16 cyfr bez spacji czy myślników">
```

- **tooLong** - zwróci true jeśli pole przekroczy wartość atrybutu **maxLength**.
- **rangeUnderflow** - dla pól które mają wartość **mniejszą** niż podaną w atrybucie **min**, zwraca true.
- **rangeOverflow** - dla pól które mają wartość **większą** niż podaną w atrybucie **max**, zwraca true.
- **stepMismatch** - gwarantuje, że pole ma wartość zgodną z kombinacjami wymuszonymi przez **min**, **max** i **step**.
- **customError** - zwraca błąd ustawiony **programistycznie** wraz ze stosownym komunikatem.

# Przydatne funkcje / zdarzenia przy walidacji

- ▶ **checkValidity()**

Wywołuje programistycznie walidację formularza. Domyślnie jest wywoływana po jego wysłaniu przez użytkownika.

- ▶ Obsługiwanie „błędnych” zdarzeń:

```
<script type="text/javascript">
// obslugiwanie blednych zdarzen
function invalidHandler(evt) {
    var poprawnosc = evt.srcElement.validity;
    // sprawdz czy jakis element typu required nie ma wpisanej wartosci
    if (poprawnosc.valueMissing) {
        // poinformuj uzytkownika, ze cos jest nie tak
    }
    // wylacz obsluge bledow wbudowana w przegladarke
    evt.preventDefault();
}
// nasluchuj blednych zdarzen
myField.addEventListener("invalid", invalidHandler, false);
</script>
```

# Stylowanie elementów

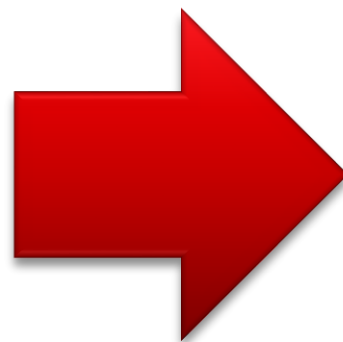
Klasa	Znaczenie
<code>valid</code>	Jakikolwiek element, który pomyślnie przeszedł walidację.
<code>invalid</code>	Wszystkie elementy, które nie przeszły walidacji i uniemożliwiają wysłanie formularza.
<code>in-range</code>	Wszystkie elementy, których wartość jest między <b>min</b> i <b>max</b> .
<code>out-of-range</code>	Wszystkie elementy, których wartość znajduje się poza zakresem.
<code>required</code>	Elementy, których wypełnienie jest wymagane.
<code>optional</code>	Wszystkie elementy, które nie są zaznaczone jako <b>required</b> .

```
:invalid {  
    background-color: red;  
}
```

# Formularz z walidacją

# LIVE

# DEMO



Zobacz plik  
Formularz z  
walidacja.html

Najlepiej działa na  
przeglądarce Google Chrome



# Ćwiczenia

- ▶ Zaprojektuj formularz do rejestracji nowego użytkownika do bloga.
  - ▶ Staraj się użyć jak najwięcej nowych komponentów.
  - ▶ Zaprogramuj jak najbardziej szczerłą walidację.
  - ▶ Wykorzystaj arkusze stylów css, aby ulepszyć wygląd formularza.
- 