jQuery

Metody i właściwości do pracy z DOM

Atrybuty - metoda addClass() i removeClass()

$("p:nth-child(even)").addClass("blue") - dodanie do wszystkich pasujących elementów klasy.

$("p:nth-child(odd)").addClass("blue title") - dodanie do wszystkich pasujących elementów dwóch klasy.

$("*:nth-child(even)").removeClass("blue") - usuniecie klasy ze wszystkich pasujących elementów (jeśli nie ma elementów lub wskazanych klas w elementach, to nic się nie dzieje). Można wskazać więcej niż jedną klasę.

Atrybuty - metoda addClass() i removeClass()

$("p:nth-child(even)").addClass("blue") - dodanie do wszystkich pasujących elementów klasy. Nie nadpisuje istniejących klas.

$("p:nth-child(odd)").addClass("blue title") - dodanie do wszystkich pasujących elementów dwóch klasy.

$("*:nth-child(even)").removeClass("blue") - usuniecie klasy ze wszystkich pasujących elementów (jeśli nie ma elementów lub klas to nic się nie dzieje). Można wskazać więcej niż jedną klasę.

Sprawdzenie czy ma klasę

```
$("p").hasClass("blue")
-> true/false

if ($("div").hasClass("first")) {
... zrób coś
}
```

Style css i jQuery

obiekt jQuery posiada metodę .css

W ten sposób możemy otrzymać informację o właściwości css pierwszego znalezionego elementu.

```
$("div").css("font-size") // zwraca np "18px"
$("div").css("font-family") // zwraca np. "Ubuntu, sans-serif"
```

metoda css()

nie tylko pobrać, ale i ustawić.

Z tym, że dla wszystkich pasujących elementów

```
$('html').css('font-size')
$("div").css("font-family", "roboto, 'open sans' ")
```

Metoda przyjmuje wtedy dwa argumenty, nazwę właściwości i wartość. Obie są stringami czyli pisane w cudzysłowach. Jeśli istniej potrzeba cudzysłowa w środku to można użyć pojedynczego.

metoda css()

$("div").css("font-size", "28px")

Zwróc uwagę, że style modyfikowane za pomocą metody css polegaj na dodaniu atrybutu style w elemencie html a więc są dodawane liniowo.

$("div").css("font-size")

Pobranie elementu natomiast dotyczy tej właściwości, która jest stosowane przez przeglądarkę (i to bardzo dobrze)

div.style.fontSize - (czysty JS) zawiera informację o wartości przechowywanej liniowo (czyli w atrybucie style elementu, a najczęściej tam nic nie ma).

metoda css() przekazanie obiekty

$("div").css("font-size", "28px") - gdy przekazujemy pojedyńczą parę właściwość - wartość, to ok.

Gdy chcemy przekazać więcej możemy użyć literału obiektu

{ }

metoda css() przekazanie obiekty

$("div").css("font-size", "28px") - wersja z przekazaniem dwóch argumentów, to zawsze jedna właściwość CSS. Ale można przekazać więcej.

Wersja z przekazaniem obiektu

```
$("#first").css({
        "font-size" : "12px",
        "color" : "#333"
})
```

metoda css() wykorzystanie operatorów

```
$("div").css("font-size", "+=28");
$("div").css("margin", "-=5");
```

Nie piszemy pikseli, ale tylko takich wartości może to dotyczyć.
Pobiera aktualną wartość i zwiększa ją.

JavaScript to jQuery

$(document.querySelector("p")) - pobieramy element i zamieniamy go na obiekt jQuery (listę zawierajaca jeden element)

---

var div = document.querySelectorAll("div") //pobieramy do listy węzłów wszystkie elementy div

var $div = $(div); zamieniali listę węzłów na obiekt jQuery.

jQuery do JavaScript

$("p")[0] - pobiera pierwszy element z listy obiketu jQuery. Uzyskujemy w ten sposób węzeł elementu (czysty obiekt JS)

$("p").get(0) - pobiera pierwszy element (metoda get jest alternatywną dla notacji z nawiasami kwadratowymi)

$("p").toArray() - zwraca tablicę (czysty obiekt JS) zawierającą wszystkie pasujące elementy. Możemy wtedy zrobić tradycyjną pętle for.

EVENTY

Bardzo zbliżone do JavaScript

metoda on()

```
$("p").on("click", function() {
        ... co ma wykonać
});
```

Zrób coś na kliknięty elemencie

```
$("p").on("click", function() {
        $(this).toggleClass("active");
}
```

//sprawdź czy kliknięty element ma klasę "active". Jeśli nie, to dodaj ją, jeśli ma, to odejmij.

this a $(this)

```
$("p").on("click", function() {
        $(this).toggleClass("active");
}
```

ale...

```
$("p").on("click", function() {
        this.toggleClass("active"); //zle
        this.classList.toggle("active") //dobrze
}
```

.on("event", funkcja)

choć często zobaczymy też zamiast on, konkretną metodę.

```
$("button.send").click(function() {})
$("button.send").on("click", function() {})  /rekomendowane
```

W jednej metodzie można wskazać kilka triggerów (wyzwalaczy)

#(input).on("keydown click", function() {})

// w tym wypadku uruchomimy funkcje zwrotną gdy ktoś kliknie lub naciśnie klawisz.

.off("event") - wyłączenie

$('p').off("click mouseover")

- ze wszystkich pasujących elementów (w tym wypadku <p>) "wyłączmy" nasłuchiwanie na click i mouseover.

Przykładowa animacja

```
$(document).on("click", function() {
        $("p").animate({
                "font-size" : "200%",
                fontSize: "200%",
        }, 2000);
});
```

Wielkości elementów

Metody pracują na pierwszym pasującym elemencie

```
$("p").width();
$("div.red").height();
```

Zwraca wielkość pudełka, ale bez marginesów, bordera i paddingu (nawet jeśli box-sizing: border-box, bo ta właściwość nie ma tu znaczenia)

Wielkości elementów

Metody pracują na pierwszym pasującym elemencie

```
$("p").innerWidth();
$("div.red").innerHeight();
```

Zwraca wielkość pudełka z paddingiem (bez bordera)

Wielkości elementów

Metody pracują na pierwszym pasującym elemencie

```
$("p").outerWidth();
$("div.red").outerHeight();
```

Zwraca wielkość pudełka z paddingiem z borderem (ale bez marginesu)

Wielkości elementów

Metody pracują na pierwszym pasującym elemencie

```
$("p").outerWidth(true);
$("div.red").outerHeight(true);
```

Zwraca wielkość pudełka z paddingiem z borderem i z marginesem

$("p").width(wartość)

$('p').width("70%"); -> style="width:70%"
$('p').width("50px"); -> style="width:50px"
$('p').width(50); -> style="width:50px"
$('p').width("+=30px"); -> pobiera aktualną i zwiększa o 30px

$("p").width(wartość)

$('p').width("70%"); -> style="width:70%"
$('p').width("50px"); -> style="width:50px"
$('p').width(50); -> style="width:50px"
$('p').width("+=30px"); -> pobiera aktualną i zwiększa o 30px

$("p").innerWidth(wartość)

$("p").innerWidth("300px");

//sprawdzi jaki jest padding np. 100px z dwóch stron (więc 200px) i ustali że width musi mieć 100px. W istocie zmienia więc style="width:100px".

okno i dokument

$(document).width() -> zwraca cały dokument czyli body z ewentualnym borderem, paddingiem i marginesem.

$(window).width() -> zwraca wielkość okna przeglądarki

//w obu przypadkach bez pasków przewijania

to samo z height()

Położenie na stronie

$(document).scrollTop()

informacja o pasku przewijania. Jeśli strona jest na górze (nie jest przewinięta) to zwróci 0 (typ number)

$(document).scrollLeft()

Można też pobrać informację o pasku poziomy i przewinieciu w lewo.

Położenie na stronie

$(document).scrollTop(200)

Jeśli podamy wartość **to przewinie w to miejsce.**

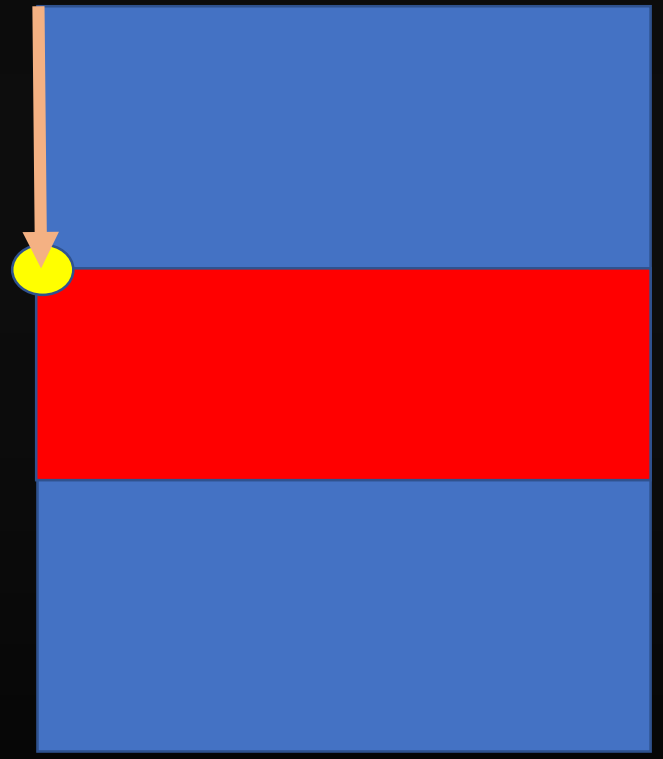
Metody .offset()

$("section.contact").offset()

Zwraca obiekt który ma dwie właściwoście

- top - jak daleko od krawędzi górnej
- left - jak daleko od krawędzie lewej

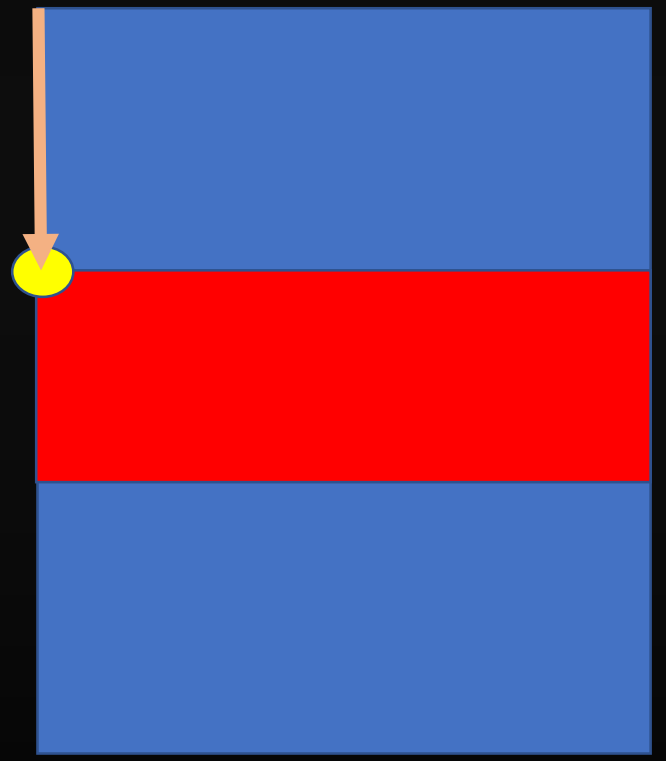
W przykłdzie top powiedzmy 200px a left:0



Metody .offset()

```
$("section.contact").offset().top
```

W przykładzie top, to powiedzmy 200px a left:0



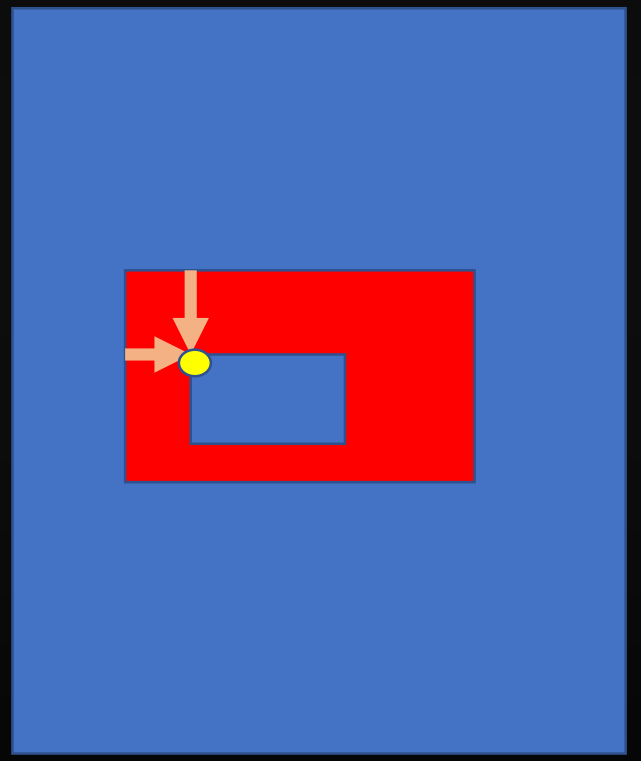
Metody .position()

$("div").position()

Relacja do obiektu nadrzędnego z absolute, fixed lub relative

- top - jak daleko od krawędzi górnej
- left - jak daleko od krawędzie lewej

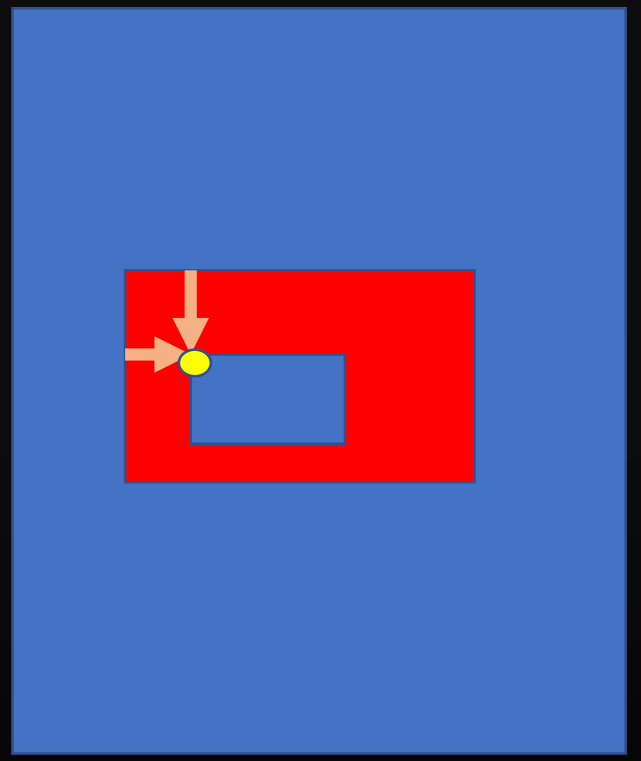
W przykłdzie top powiedzmy 25px top o 15px left/



Metody .position() i .offset() w praktyce

```
$("div").position().left
$("div").position().top

$("div").offset().top
$("div").offset().left
```



PROJEKTY JQUERY