Zadanie 20: DOM – kalkulator zamówienia pizzy

Treść zadania

Wykonaj stronę internetową pizzerii z kalkulatorem kosztu zamówienia, wykorzystując edytor zaznaczający składnię oraz przeglądarkę internetową. Utwórz folder o nazwie zadanie20. Po skończonej pracy wszystkie pliki zapisz w tym folderze.

Witryna internetowa

Wygląd strony po otwarciu przedstawia ilustracja 1, a po wybraniu opcji i kliknięciu przycisku OBLICZ – ilustracja 2.



Ilustracja 1. Strona pizzeria.html po otwarciu



Ilustracja 2. Strona po wybraniu rozmiaru „Średnia”, dodatków „Ser extra” i „Pieczarki”, liczby pizz 3 i kliknięciu przycisku OBLICZ

Cechy strony pizzeria.html:

- Zapisana w języku HTML5
- Zadeklarowany polski język zawartości witryny
- Jawnie zastosowany standard kodowania polskich znaków UTF-8
- Tytuł strony „Pizzeria Pod Piecem”
- Arkusz stylów w pliku o nazwie styl.css prawidłowo połączony z kodem strony
- Skrypt w pliku o nazwie skrypt.js, połączony ze stroną znacznikiem `<script>` umieszczonym na końcu bloku `<body>`
- Podział strony na bloki: baner, poniżej menu, poniżej blok lewy i blok prawy, na dole stopka. Podział zrealizowany za pomocą znaczników `<header>`, `<nav>`, `<section>` (identyfikator `lewy`), `<aside>` (identyfikator `prawy`) i `<footer>` tak, aby układ bloków był zgodny z ilustracją 1
- **Zawartość bloku banera:** nagłówek pierwszego stopnia o treści „Pizzeria Pod Piecem”
- **Zawartość bloku menu:** trzy odnośniki (`<a>`) o treści „Strona główna”, „Menu” i „Kontakt”, prowadzące do adresu #
- **Zawartość bloku lewego:** nagłówek drugiego stopnia „Zamów pizzę” oraz:
- podpis „Rozmiar:” (`<label>` powiązany atrybutem `for`, klasa `etykieta`) i lista rozwijana (`id="rozmiar"`, klasa `pole`) z elementami „Mała” (`value="20"`), „Średnia” (`value="28"`), „Duża” (`value="36"`)

- paragraf „Dodatki:” (klasa `etykieta`) oraz trzy pola wyboru (`type="checkbox"`, `name="dodatek"`), każde w znaczniku `<label>` klasy `dodatek`: „Ser extra” (`value="4"`), „Szynka” (`value="5"`), „Pieczarki” (`value="3"`)
- podpis „Liczba pizz:” (klasa `etykieta`) i pole numeryczne (`type="number"`, `id="liczba"`, wartość początkowa 1, klasa `pole`)
- przycisk `id="oblicz"` o treści „OBLICZ” oraz pusty paragraf o identyfikatorze `wynik`
- **Zawartość bloku prawego:** nagłówek drugiego stopnia „Cennik”; lista nieuporządkowana z pozycjami: „Pizza mała – 20 zł”, „Pizza średnia – 28 zł”, „Pizza duża – 36 zł”, „Ser extra – 4 zł”, „Szynka – 5 zł”, „Pieczarki – 3 zł”; nagłówek drugiego stopnia „Twoje zamówienie” oraz pusty paragraf o identyfikatorze `opis`
- **Zawartość bloku stopki:** paragraf o treści „Stronę opracował: ”, dalej wstawiony numer zdającego, zapisany za pomocą znacznika semantycznego oznaczającego tekst uwypuklony, formatowany domyślnie jako pochylony

Styl CSS witryny internetowej

Styl CSS zdefiniowany jest w całości w zewnętrznym pliku o nazwie styl.css. Cechy formatowania CSS działające na stronie:

- Domyślnie, dla wszystkich selektorów: krój czcionki „Georgia”, w przypadku braku serif
- Dla ciała strony: kolor tła Cornsilk
- Dla banera: kolor tła DarkRed, kolor czcionki White, wyrównanie tekstu do środka, marginesy wewnętrzne 10 px
- Dla menu: kolor tła Maroon, marginesy wewnętrzne 10 px
- Dla selektora odnośnika: marginesy wewnętrzne górny i dolny 0 px, lewy i prawy 25 px, kolor czcionki Gold, brak podkreślenia (`text-decoration: none`). Gdy kursor myszy znajdzie się na odnośniku, tekst jest podkreślony
- Dla bloku lewego: model pudełkowy border-box, położenie po lewej stronie (`float: left`), szerokość 60%, wysokość 480 px, marginesy wewnętrzne 20 px, kolor tła Ivory
- Dla bloku prawego: model pudełkowy border-box, położenie po lewej stronie (`float: left`), szerokość 40%, wysokość 480 px, marginesy wewnętrzne 20 px, kolor tła Wheat
- Dla nagłówka drugiego stopnia: kolor czcionki DarkRed
- Dla klasy etykieta: wyświetlanie blokowe, marginesy zewnętrzne górny 12 px, prawy 0, dolny 4 px, lewy 0, czcionka pogrubiona
- Dla klasy dodatek: wyświetlanie blokowe
- Dla klasy pole: model pudełkowy border-box, szerokość 100%, marginesy wewnętrzne 8 px, obramowanie linią ciągłą o grubości 1 px i kolorze Gray, zaokrąglenie rogów 5 px
- Dla przycisku: marginesy zewnętrzne górny 15 px, marginesy wewnętrzne 10 px (góra i dół) i 30 px (lewy i prawy), kolor tła DarkRed, kolor czcionki White, brak obramowania, zaokrąglenie rogów 20 px, kursor pointer. Gdy kursor znajdzie się na przycisku, kolor tła zmienia się na FireBrick
- Dla elementu o identyfikatorze wynik: kolor czcionki DarkRed, rozmiar czcionki 130%, czcionka pogrubiona
- Dla elementu o identyfikatorze opis: czcionka pochylona
- Dla stopki: zakończenie opływania (`clear: both`), kolor tła DarkRed, kolor czcionki White, wyrównanie tekstu do środka, marginesy wewnętrzne 1 px

Uwaga: nie należy zmieniać nazw klas ani identyfikatorów, gdyż jest to uwarunkowane projektem całej witryny.

Skrypt działający po stronie klienta

Wymagania dotyczące skryptu:

- Zapisany w języku JavaScript w pliku skrypt.js, stosuje znaczące nazewnictwo zmiennych w języku polskim lub angielskim
- Skrypt uruchamia się po kliknięciu przycisku OBLICZ i pobiera: wybrany rozmiar pizzy (wartość listy), zaznaczone dodatki oraz liczbę pizz
- Koszt zamówienia to (cena rozmiaru + suma cen zaznaczonych dodatków) pomnożone przez liczbę pizz
- Skrypt wyświetla w paragrafie o identyfikatorze wynik tekst według wzoru „Koszt zamówienia: <wartość> zł”
- W paragrafie o identyfikatorze opis skrypt wyświetla tekst według wzoru „Rozmiar: <nazwa rozmiaru>, dodatków: <liczba zaznaczonych dodatków> (razem <suma cen dodatków> zł), liczba pizz: <liczba pizz>”
- Gdy liczba pizz nie jest liczbą całkowitą większą od 0, skrypt wyświetla w paragrafie wynik tekst „Podaj poprawną liczbę pizz.”, a paragraf opis jest pusty

Tabela 1. Wybrane pola i metody modelu DOM języka JavaScript

Wyszukiwanie i zmiana elementów	Dodatkowo dopuszczalne
`document.getElementById(id)`	`element.options[element.selectedIndex].text`
`document.getElementsByName(ElementName)`	`element.checked`
`element.innerHTML = "wartość"`	`Number(tekst), Number.isInteger(liczba)`
`onclick`	`element.value`

UWAGA: po zakończeniu pracy sprawdź działanie strony w przeglądarce. W folderze powinny znajdować się pliki: pizzeria.html, styl.css, skrypt.js.
Ocenie podlegać będą 3 rezultaty: zawartość witryny internetowej, styl CSS witryny internetowej, skrypt.

Rozwiązanie

Pytania teoretyczne

Pytanie 1. Co wypisze `console.log("28" + 4)`?

A. 32

B. 284 (tekst)

C. NaN

D. błąd

Pytanie 2. Czym różni się `rozmiar.value` od `rozmiar.options[rozmiar.selectedIndex].text` dla listy rozwijanej? Do czego użyto każdego z nich w skrypcie?

Pytanie 3. Dlaczego do sprawdzenia, czy dodatek jest wybrany, używamy właściwości `checked`, a nie `value`?

Pytanie 4. Co zwrócą `Number("")` i `Number("2,5")`? Dlaczego warunek `liczba < 1` nie wystarcza do walidacji liczby pizz?

Pytanie 5. Co zwraca `document.getElementsByName("dodatek")` i dlaczego ta metoda jest wygodna dla grupy pól wyboru?

Odpowiedzi do pytań teoretycznych