

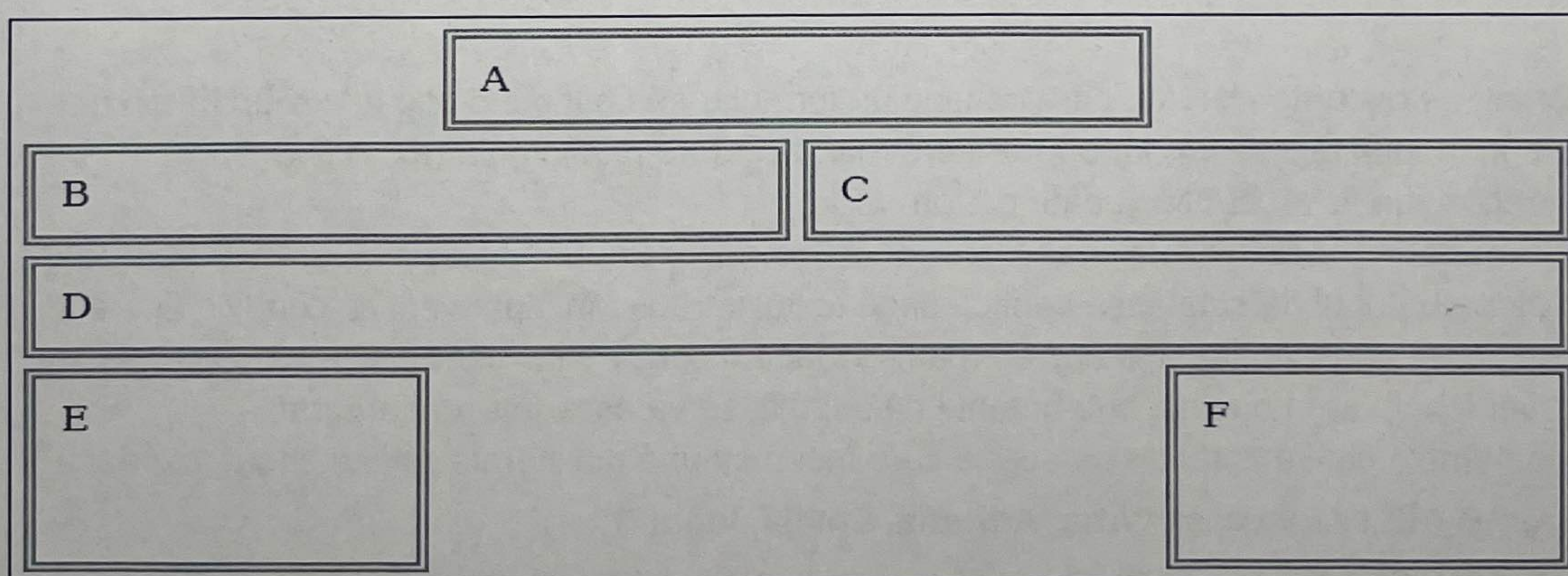
## Examen Tehnici Web – 30 ianuarie 2025

### Regulament:

- fiecare subiect se va rezolva într-un fisier separat cu extensia html (ex: p1.html, p2.html, p3.html, p4.html), codurile CSS și JavaScript corespunzătoare fiecărui exercițiu vor fi scrise fie în fișiere separate, fie direct în fișierul html.
- aveți 2 ore pentru rezolvarea subiectelor și încă 5 minute pentru trimiterea soluțiilor.
- rezolvările se vor încărca sub forma unei arhive (.zip), a cărei nume va fi de forma **nrgrupa\_nume\_prenume**, folosind următorul link: <https://tinyurl.com/yc4tehej>
- fiecare exercițiu are **1,5 puncte**.
- condiția de promovare a testului: **minim 3 puncte**.
- în timpul examenului, **NU** aveți voie cu nici o aplicație/mediu de comunicare/mobil, cum ar fi facebook, skype, alte tipuri de messenger sau aplicații ale rețelelor de socializare. Nu aveți voie să comunicați cu colegii.
- **Important!** Orice similaritate a codului se pedepsește cu anularea punctajului subiectului/subiectelor pentru toți cei implicați.

**P1.** Creați în HTML un element div cu clasa "box" în care se vor găsi alte 6 divuri (copii direcți).

- Scrieți o regulă CSS astfel încât divul cu clasa "box" să aibă lățimea egală cu 2/3 din lățimea ferestrei, padding de 5px și un border ales de voi. Adăugați reguli CSS astfel încât pagina să arate ca în imaginea de mai jos și să aveți: div-urile cu un index multiplu de 3 să aibă background galben și culoarea textului neagră iar celelalte div-uri să aibă background gri și culoarea textului albă. Toate divurile copii vor avea un padding de 10px și un border roșu dublu de 5px. Între elementele din container să existe un spațiu de 5px atât pe linii cât și pe coloane.



- Scrieți un media query pentru lățimea ferestrei între 250px și 650px cu efectele:

- a) divurile copii să se afișeze unul sub altul, întinse pe toată lățimea divului container.
- b) când aducem cursorul pe divul părinte, acesta să își dubleze înălțimea iar ultimul element din div să își schimbe culoarea borderului în albastru, modificarea proprietăților se va face lent timp de 3 secunde.



**P2.** Creați un document HTML fără conținut (nu conține alte elemente în interiorul body-ului). Definiți clasele CSS *"color1"*, *"color2"*, *"color3"*, *"color4"*, *"color5"* cu culori de background diferite. Scrieți cod JavaScript pentru a rezolva următoarele:

- a) La încărcarea paginii, fiecare apăsare a tastei **"b"** va avea ca efect crearea unui buton (element html button) de dimensiuni 100px x 50px. Fiecare buton va primi la creare o clasă aleatoare din mulțimea claselor *{"color1", "color2", "color3", "color4", "color5"}*. Afișați culoarea butonului ca text pe acel buton (nu contează formatul culorii).
- b) La click oriunde în fereastră (în afara butoanelor) se va afișa într-un alert numărul de butoane existente în pagină. La dublu click pe oricare dintre butoane, toate butoanele care au aceeași clasă cu butonul pe care s-a dat dublu click vor deveni inactive, însă nu mai apare alertul. Dacă se apasă tasta **"Backspace"**, toate butoanele inactive vor fi șterse.

**P3.** Creați un document HTML care conține un element checkbox cu eticheta "pauză" care nu este bifat și o listă neordonată (element ul) cu 7 elemente având ca texte zilele săptămânii: "luni", "marți", ..., "duminică". Scrieți cod JavaScript pentru a rezolva:

- a) La încărcarea paginii, la fiecare 3s, ultimul element al listei se va muta la începutul listei devenind primul element al listei. La bifarea checkbox-ului mutarea se va opri; la debifarea checkbox-ului mutarea va porni din nou.
- b) După ce checkbox-ul a fost bifat, la reîncărcarea paginii, elementele listei să apară în ordinea în care erau în momentul bifării checkbox-ului (folosiți *localStorage* pentru păstrarea ordinii).

**P4.** Creați un document HTML care conține un formular cu un input de tip text și un buton de submit. În inputul text se vor introduce litere mari separate prin virgulă (ex: A,B,C,E).

Considerăm un fișier JSON cu date de forma:

```
{ "persoane": ["Ana", "Adriana", "Mihai", "Corina", "Emilia", "Maria", "Ionut", "Marius"] }
```

La click pe butonul de submit se verifică dacă textul introdus în input verifică condiția (litere mari separate prin virgulă) și în caz afirmativ se face o cerere Ajax către fișierul JSON (aflat pe un server local) iar în pagină, sub butonul de submit, se va afișa într-un paragraf:

- toate numele din vectorul de persoane care încep cu una din literele citite în input (ex: dacă se introduc **A,E,I** se va afișa **"Ana, Adriana, Emilia, Ionut"**).
- **"Nu exista persoane"**, dacă nu există nici un nume în vectorul de persoane care începe cu una din literele menționate.

Dacă textul introdus în input nu satisface condiția cerută (să existe litere mari separate prin virgulă) se va afișa într-un alert **"Date invalide"** iar cererea nu va mai fi trimisă.