

După cum am menționat în lecțiile anterioare, în HTML există elemente care definesc scopul și intenția elementelor unei pagini HTML, care se bazează pe elemente semantice. În această lecție vom analiza în detaliu care este funcția lor.

Tag-urile pe care le prezentăm în tabelul de mai jos deja le folosim și este specificat că pentru browser au o semnificație aparte. Un exemplu de semnificație poate fi pentru tag-urile heading, em și strong, și chiar dacă nouă ni se par la fel vizual, dacă folosim alternativele lor pentru browser, acestea sunt importante, deoarece spun dacă conținutul este important, care titlu se folosește pentru o anumită parte din text etc. Toate acestea permit motoarelor de căutare o căutare și o prezentare mai simple a conținuturilor relative.

Tag-ul HTML		Prezentare vizuală	Semnificație pentru browser
	Text italic	Parte mai importantă a textului din pagină	
	Text îngroșat	Parte importantă a textului din pagină	
<h1>	Titlul de primul nivel	Text care oferă informație la motorul de căutare despre ce să aștepte pe pagină după titlul documentului	
<h2>	Titlul de nivelul doi	Text pe care motorul de căutare îl ia ca o descriere a părților din conținut pe pagină după subtitluri	

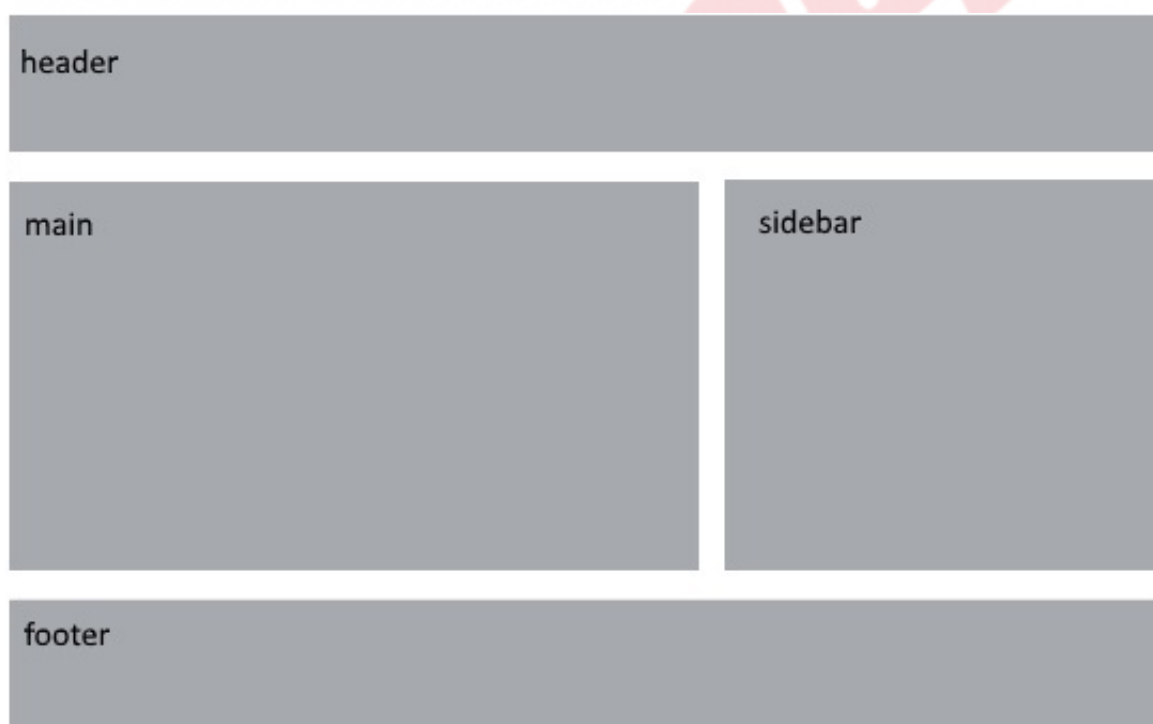
Tabelul 13.1. Elemente semantice

Prin adăugarea tag-urilor semantice în document, precum cele de mai sus și multe altele, noi oferim informații suplimentare legate de documentele noastre. Mai exact, tag-urile semantice aduc la cunoștință browserului care este semnificația site-ului nostru și ce fel de conținuturi are. O pagină bine scrisă din punct de vedere semantic se găsește mai ușor în motorul de căutare, ceea ce simultan și crește traficul, adică numărul utilizatorilor de pe site.

În standardul HTML5 avem și noi elemente semantice care definesc

părți din pagină și oferă informații suplimentare despre ele. Aceste elemente noi, pe care le putem folosi acum, se comportă ca și elemente div și între ele nu există niciun fel de diferență. Totuși, pe de altă parte, acestea contribuie la semantica paginii, respectiv definesc scopul și intenția părților paginii. Browserele, roboții motoarelor de căutare (Google, Bing) și diferite alte software-e le pot analiza și, în funcție de scop, să le trateze altfel.

După cum am văzut și în lecțiile precedente, acesta este un layout frecvent al unei pagini moderne:



Imaginea 13.1. Exemplu de layout modern al paginii web

Fiecare dreptunghi reprezintă un div, o unitate cu care grupăm conținutul.

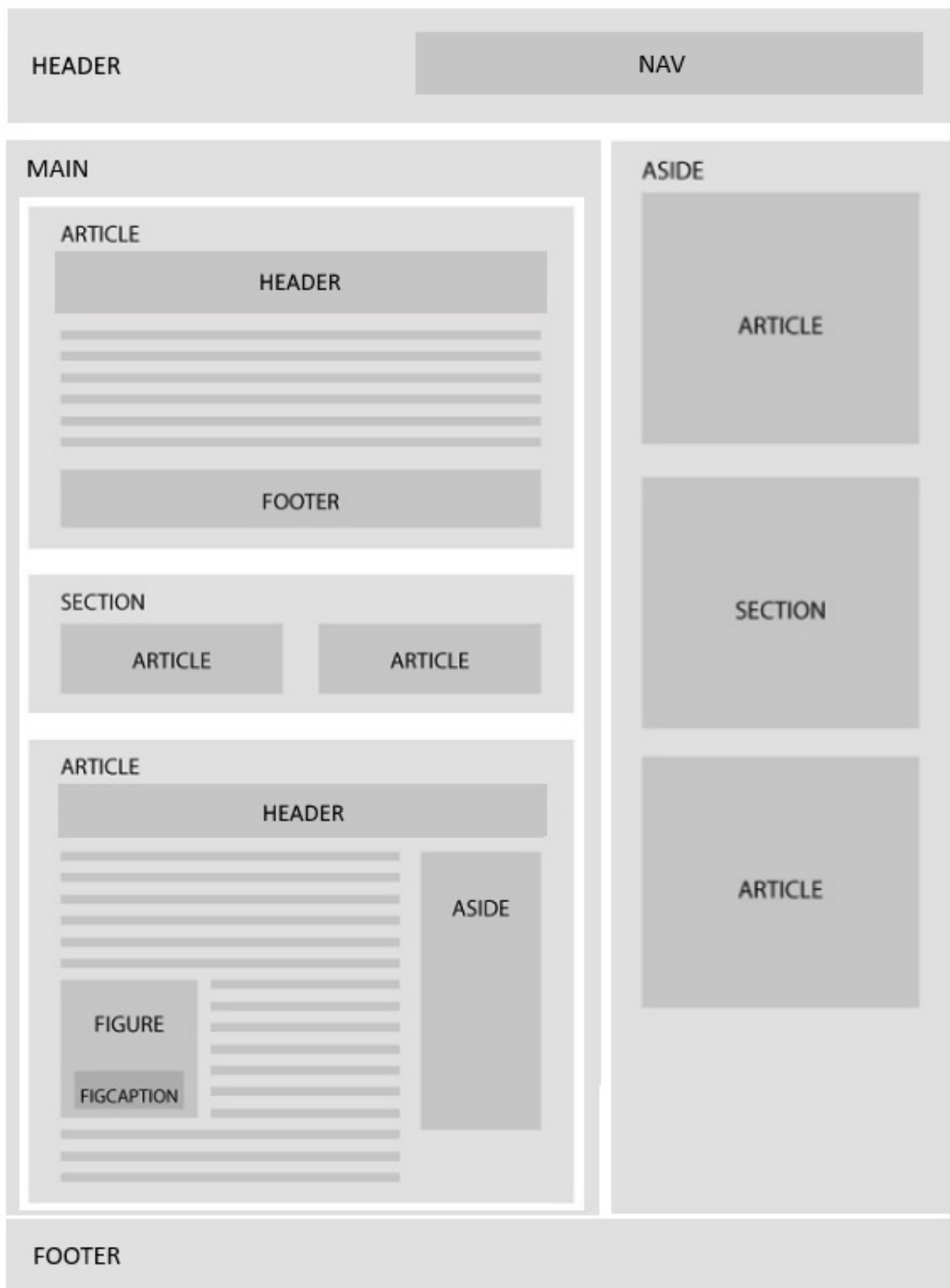
În general, pentru fiecare parte a paginii se pune un element div separat, iar apoi, cu alte div-uri în ele sunt separate anumite unități.

Atunci când definim un astfel de layout cu ajutorul elementelor div, noi

construim elementele paginii, dar motorul de căutare nu este conștient de aceste părți, deoarece doar pentru noi dăm nume la id-uri sau la clasele acestor elemente div, ca de exemplu header, footer, main etc. Dacă folosim elemente semantice în locul anumitor elemente div, îi oferim motorului de căutare și informații suplimentare despre aceste elemente.

Hai deți să vedem asta într-un exemplu. Luăm același layout, dar de data asta părțile sale le vom defini cu ajutorul elementelor semantice, pentru a obține ceva precum:

LINKgroup



Imaginea 13.2. Exemplu de layout modern al paginii web folosind

elemente semantice

Puteți să recunoașteți unele elemente care au rămas neschimbate, precum Header, Footer și Main. Din punct de vedere semantic au același nume, doar tag-ul cu care le înregistrăm în HTML este diferit. Există mai multe elemente noi care se repetă pe pagină și pentru layout-ul div acestea ar fi doar elemente care în continuare împart main în unități etc. Acum vom explica pe rând, separat, fiecare element din imagine.

Notă:

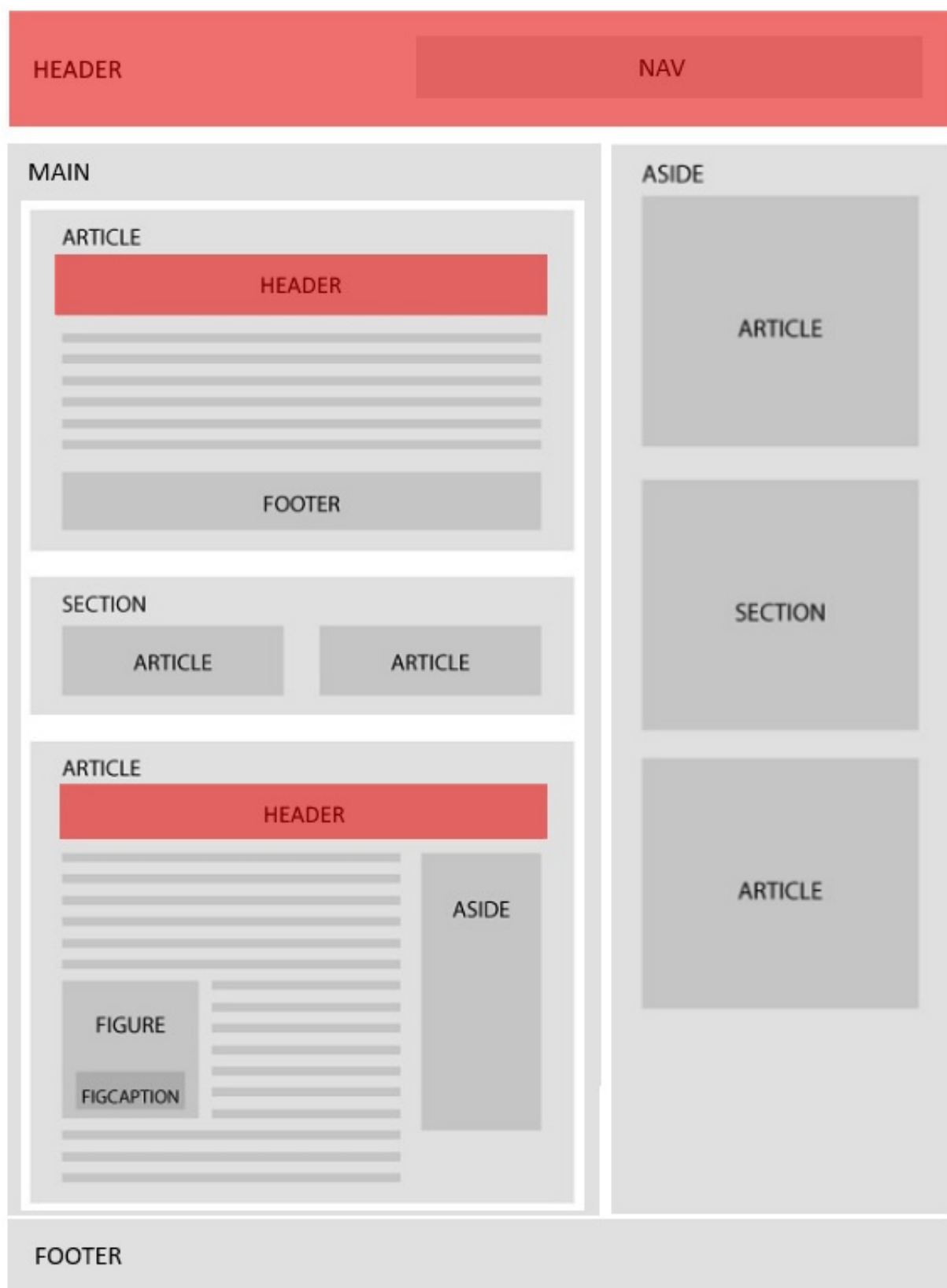
Înainte de a începe, vreau să menționez următoarele: când definim elementele semantice ale paginii, acest lucru înseamnă că ele vor primi automat și stilizarea CSS necesară. De exemplu, dacă definim elementul `<header>`, acesta nu va fi automat în vârful paginii, pe întreaga lățime, ci, ca și la elementul div, trebuie să facem stilizarea prin CSS.

Elementul header - `<header>`

`<header> </header>`

Acest element se folosește pentru definirea header-ului individual a unei anumite părți/a elementului paginii (de exemplu `<article>` sau `<section>`) sau mult mai des pentru a defini header-ului principal al paginii, dacă în partea de sus a paginii se separă antetul acestei părți și titlul, dacă trebuie etc.

Dacă se pune direct pe pagină header-ul site-ului, acesta conține de cele mai multe ori logo-ul, căutarea, navigarea etc.



Imaginea 13.3. Exemplu de aspect (layout) al unei pagini cu

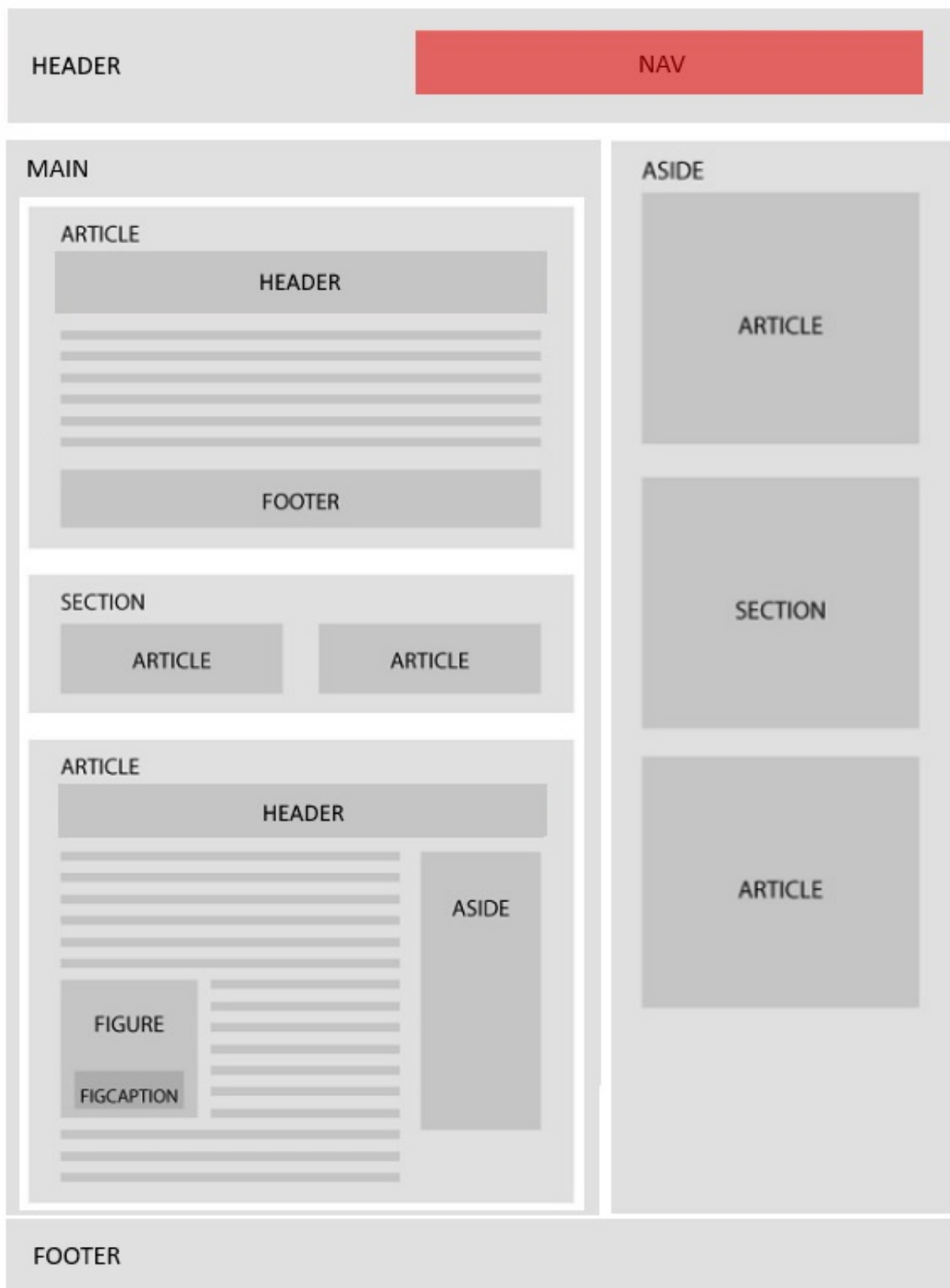
elementele header marcate

În imaginea de mai sus vedem încă o schiță a unui aspect al unei pagini. Elementele header sunt evidențiate cu culoarea roșie și vedem că, după cum am spus deja, se pot folosi pentru antetul principal al paginii, dar și pentru antetele individuale ale unor părți ale paginii, a unităților mai mici.

Navigation - <nav>

`<nav>` `</nav>`

Elementul de navigare se folosește (după W3C) pentru setarea secțiunilor de navigare principale. Cu alte cuvinte, atunci când trebuie să grupăm linkurile de navigare într-un element, folosim `<nav>`.



Imaginea 13.4. Exemplu de aspect (layout) al unei pagini cu elementul

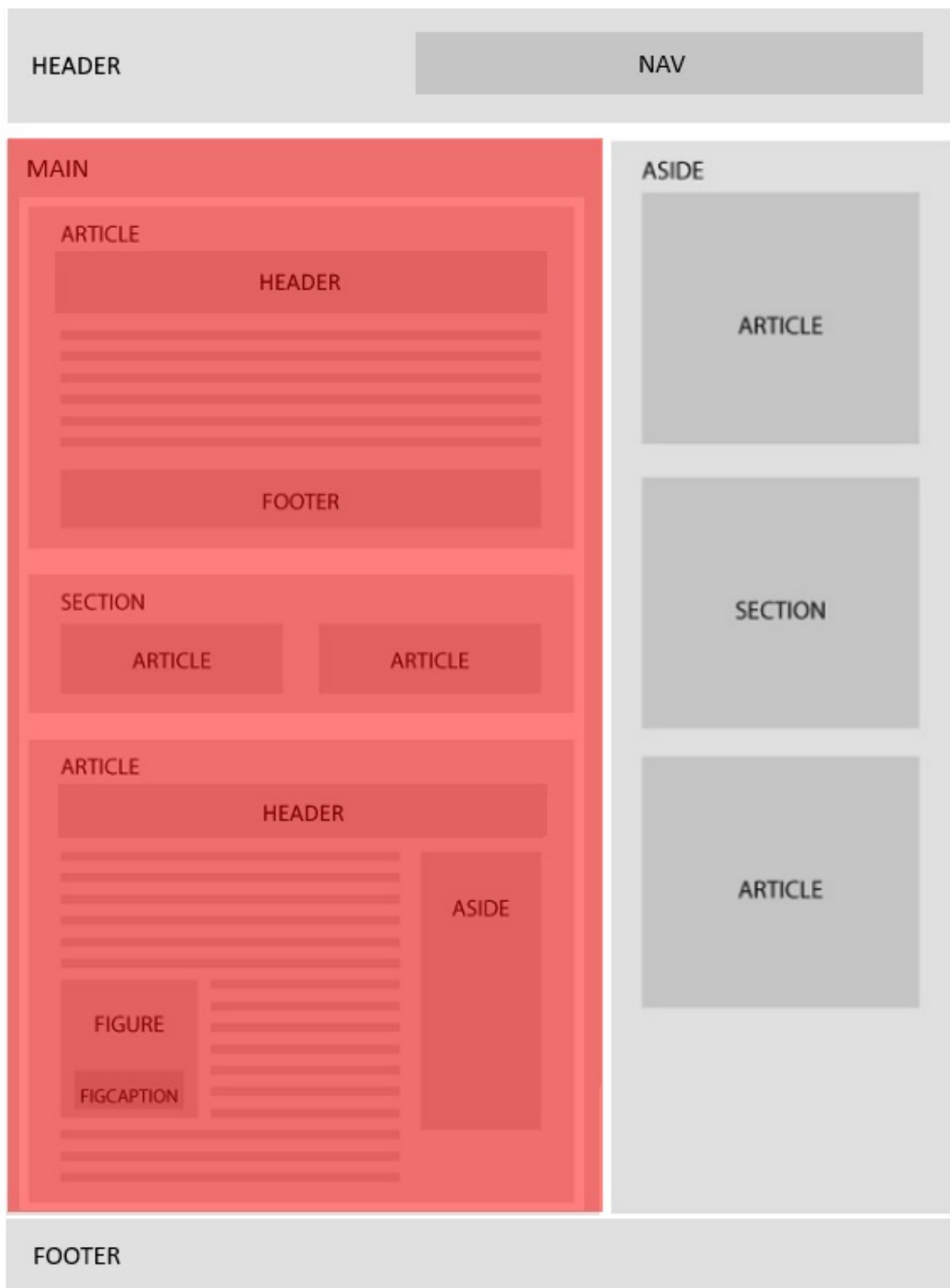
nav marcat

Încă mai există controverse și discuții privind utilizarea lui `nav` pentru toate pozițiile cu linkuri pe pagină sau doar pentru navigarea principală. O regulă (în general) acceptată spune că `nav` trebuie să se folosească pentru blocuri de navigare mai mari. De exemplu, dacă în footer punem doar linkurile pentru reguli de utilizare, informații juridice și pagina de start, acestea nu trebuie să le punem în `nav`, dar dacă în același folder avem linkurile principale complet repetate, principala navigare a site-ului, atunci trebuie să folosim `nav`.

Main - `<main>`

`<main> </main>`

Elementul `main` este unul dintre cele mai interesante elemente semantice noi și unul dintre cel mai controversat pas în dezvoltarea HTML5. Acest element este inclus după altele, deoarece multă vreme a existat întrebarea dacă trebuie să se seteze și să se creeze un element semantic nou pentru conținutul „principal”.



Imaginea 13.5. Exemplu de aspect (layout) al unei pagini cu elementul

main marcat

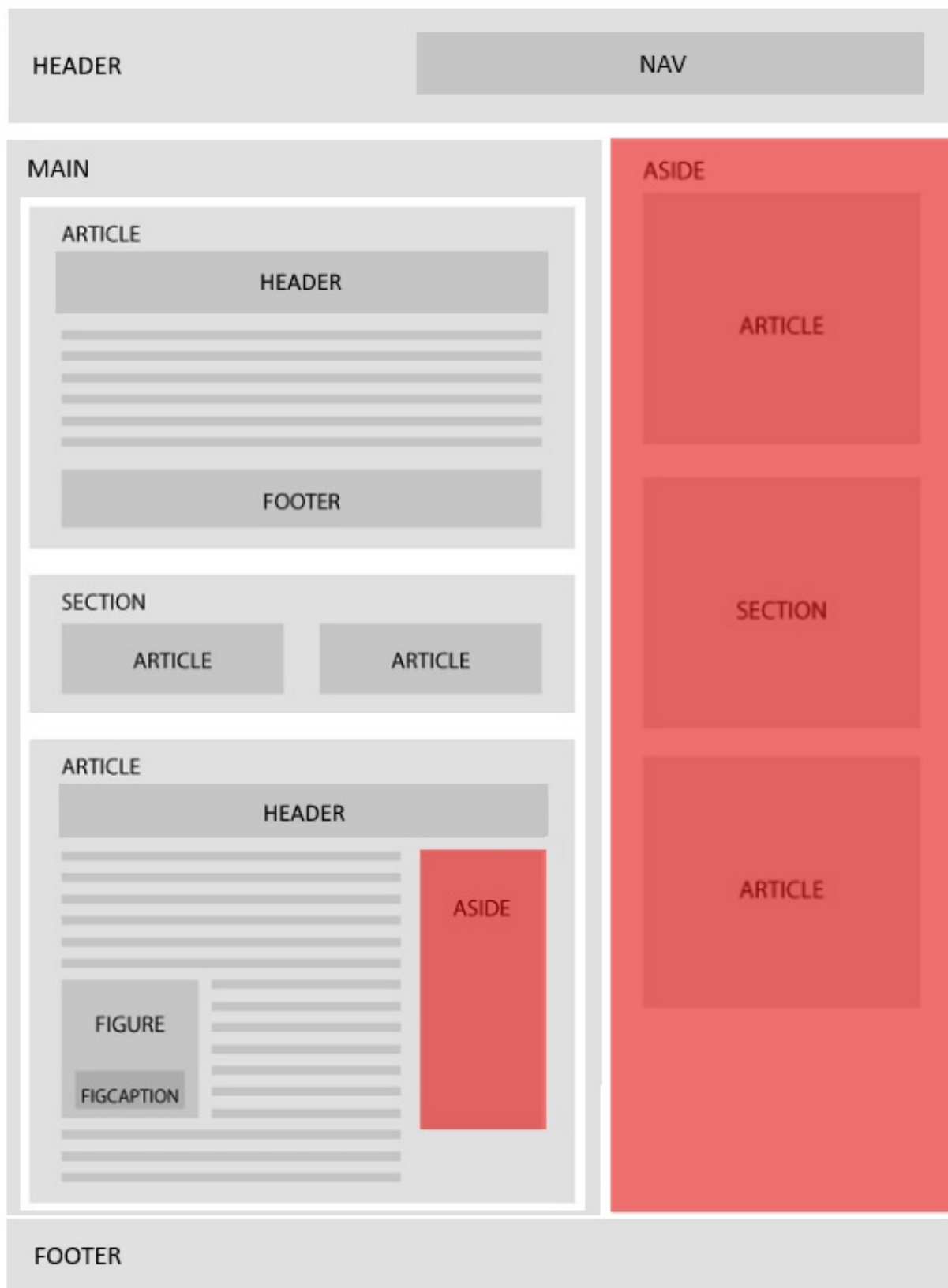
Elementul main reprezintă partea principală, centrală a paginii și nu trebuie să se găsească în cadrul elementelor <article>, <aside>, <footer>, <header> sau <nav>. **De asemenea, nu avem voie să repetăm elementul main mai mult decât o dată pe o pagină concretă.**

Astăzi, main este partea principală și se poate folosi cu regularitate, dar rămâne problema suportului în Internet Explorer. De fapt, nu toate versiunile Internet Explorer suportă și recunosc elementul main și, de aceea, se pune în discuție justificarea utilizării sale reale pe site-uri (lista completă a suportului în browsere: <http://caniuse.com/#feat=html5semantic>). Din ianuarie 2016, Microsoft a întrerupt suportul și actualizarea pentru toate versiunile mai vechi <https://www.microsoft.com/en-us/WindowsForBusiness/End-of-IE-support> și s-a focusat în totalitate pe Edge de care trebuie să aibă grijă, deoarece încă există mulți utilizatori care folosesc Internet Explorer.

Aside - <aside>

<aside> </aside>

Elementul aside se folosește pentru crearea unui conținut secundar și are două moduri de aplicare, în funcție de context, similar cu elementul header. Dacă este în cadrul elementului <article> (pe care îl vom explica mai târziu), conține informații despre el, dar nu sunt decisive, de exemplu, lista bibliografiei suplimentare pe lângă textul dat etc. Dacă este în afara elementului <article>, se referă la pagina întreagă. Deseori se folosește ca un sidebar principal. Aici se pot găsi și reclame, formulare pentru stabilirea contactului, linkuri către paginile recomandate etc.



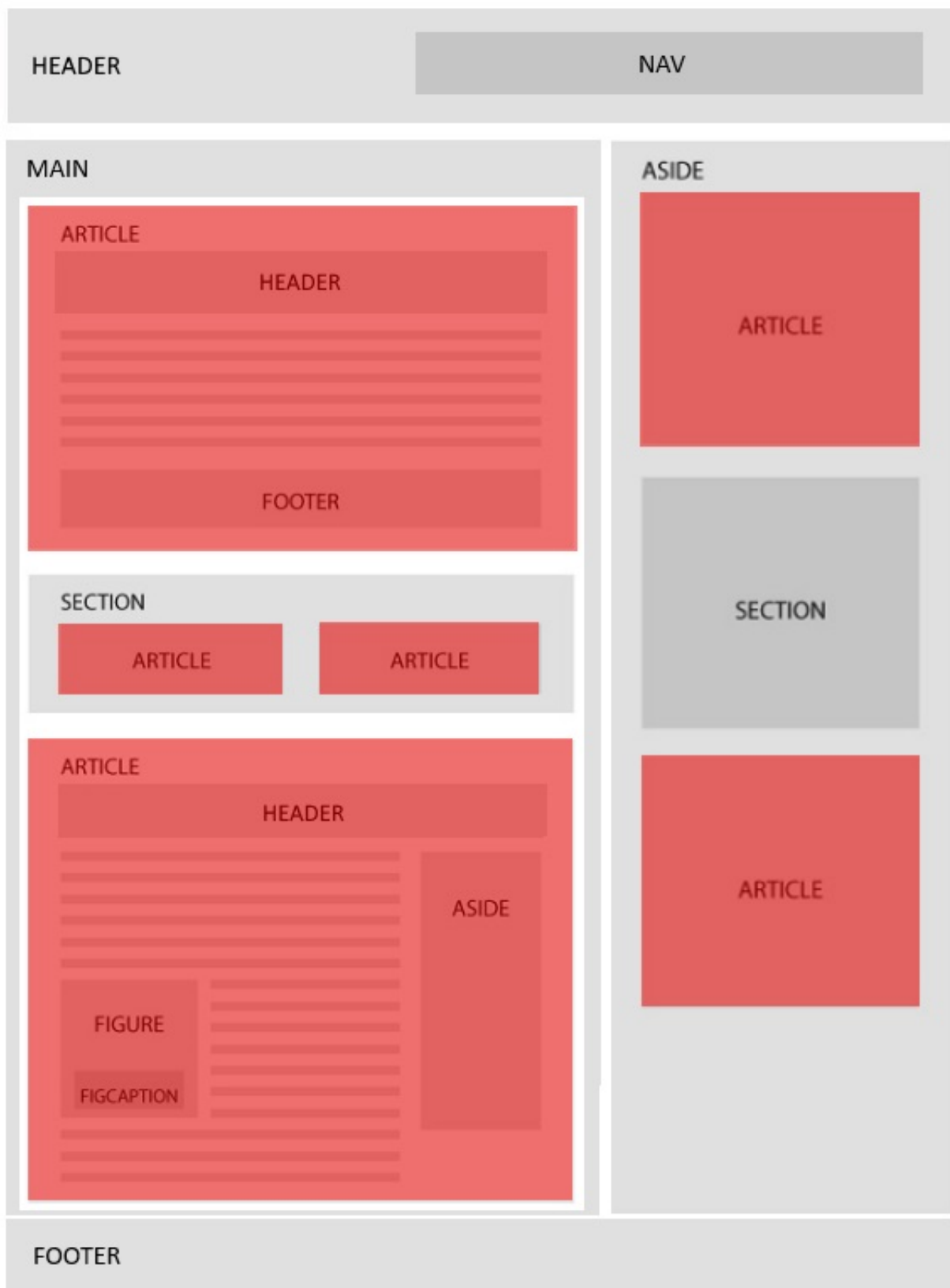
Imaginea 13. 6. Exemplu de aspect al unei pagini cu elementele aside marcate

În imaginea de mai sus vedem că principalul sidebar al site-ului este setat ca element aside care conține altele (section, article...), dar aside apare, de asemenea, și ca un conținut secundar în cadrul unui element article. Principalul sidebar se referă, la fel, la site-ul întreg, iar aside, în cadrul elementului article, se referă doar la acest article.

Article - <article>

<article> </article>

Elementul article se comportă ca un cadru pentru orice parte a paginii care poate sta independent de restul conținutului. Este destinat articolelor, introducerii pe blog, postărilor pe forum și oricărei altei părți independente a conținutului.



Imaginea 13. 7. Exemplu de aspect al unei pagini cu elementele aside

marcate

Dacă nu sunteți siguri de valabilitatea utilizării elementului `article`, ar trebui să vă puneți întrebarea dacă l-am putea copia pe o altă pagină și dacă asta ar avea atunci sens.

Deși poate fi mult mai complicat. De exemplu, dacă setăm mai multe știri [teaser](#) pe o pagină, fiecare dintre ele poate fi un `article` special.

Pe de altă parte, pe o altă pagină care conține un text complet al unei știri, putem să avem și comentarii pe lângă această știre. Întregul text cu toate comentariile poate fi cuprins de un tag `article`, iar în el comentariile pot fi tag-urile `article` individuale (vezi imaginea de mai jos).



Imaginea 13.8. Schița unei utilizări a elementului article

În imaginea prezentată mai sus vedem că o știre, respectiv un text complet cu titlu și cu comentarii, este setat în cadrul unui element article. În acest element se află direct exemplul h2 pentru titlu, câteva paragrafe de text, precum și alte elemente.

Dar, în plus, la sfârșitul elementului nostru article se află și comentarii lăsate de vizitatori. Fiecare comentariu poate fi un article nou separat, deoarece comentariul face o unitate pentru el.

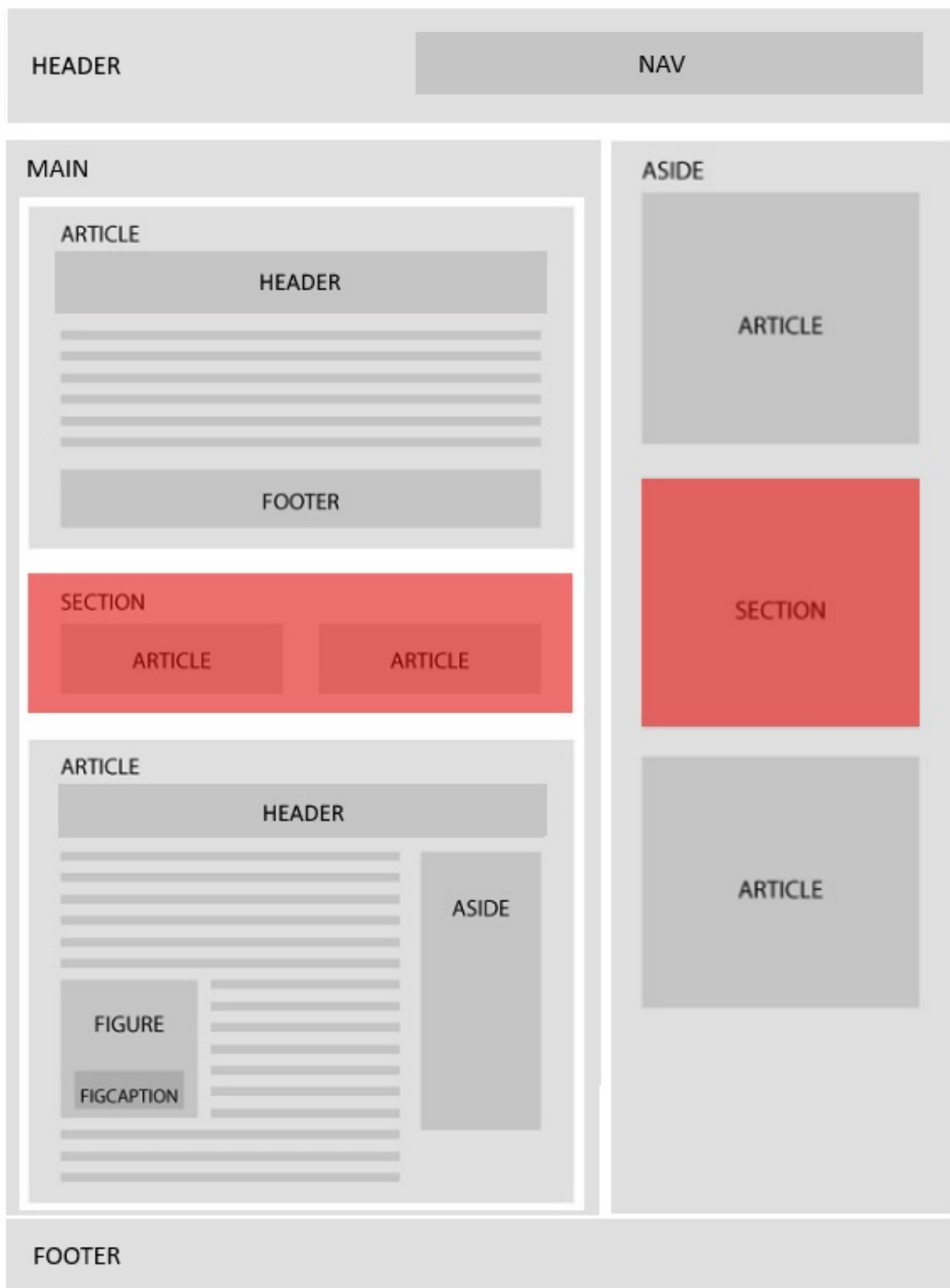
Deseori se poate să fie confuză diferența dintre elementul `<article>` și

`<section>`, pe care îl vom explica în continuare.

Section - `<section>`

`<section>` `</section>`

Elementul `section` este considerat de mulți autori drept o înlocuire pentru `div`, dar **nu** este cazul. Elementele `div` nu sunt eliminate din specificația HTML5, ci și-au păstrat rolul ca elemente generice, non-semantice, atunci când utilizarea elementelor semantice noi (care nu sunt destinate în mod specific) nu este necesară.



Imaginea 13.9. Exemplu de aspect al unei pagini cu elementele section

marcate

Section are un scop diferit față de elementul div. În general îl folosim ca să grupăm mai multe elemente, pentru a crea o secție, de exemplu, mai multe introduceri articole, și astfel, în general, întotdeauna se introduce un titlu pentru section. Dacă nu este nevoie de titlu, întrebați-vă dacă div nu este o soluție mai bună.

Exemplu:

```
<section>
```

```
<h2> Section heading </h2>
```

```
<article>...</article>
```

```
<article>...</article>
```

```
  <article>...</article>
```

```
</section>
```

Sau putem seta mai multe secții într-un element articole. De exemplu:

```
<article>
```

```
  <section>...</section>
```

```
  <section>...</section>
```

```
</article>
```

În primul exemplu am grupat mai multe elemente articole într-o unitate pe care o și denumim, iar în al doilea exemplu am împărțit un element articole în mai multe secții – împărțiți articolul în secții, dar pentru asta există utilizări reale. Întrebați-vă dacă fiecare dintre aceste secțiuni poate avea și un titlu propriu.

De asemenea, ceva mai devreme am văzut un exemplu pentru article unde comentariile sunt elemente article separate. În acest exemplu, toate aceste comentarii le putem pune într-un section tag și să punem un titlu.

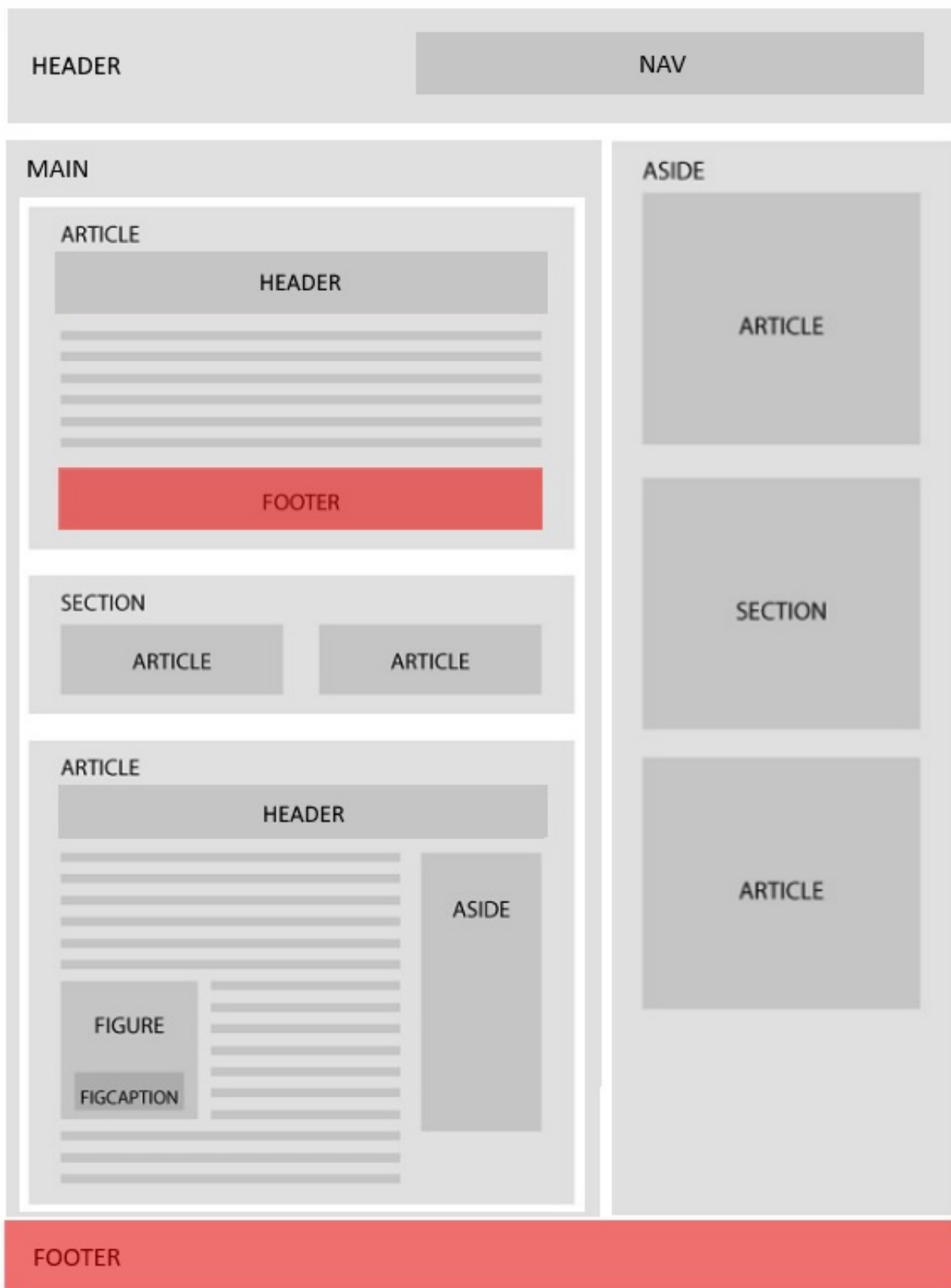
Notă:

Rețineți, dacă trebuie să puneți doar un element pentru al stiliza cu CSS, **atunci nu folosiți section**, ci div-ul.

Footer - <footer>

<footer> </footer>

Footer seamănă cu header, cu un alt scop, dacă header-ul stabilește antetul, footer-ul stabilește partea finală a elementului sau a paginii. Dacă este în cadrul paginii poate avea numele autorului de text, pagini legate etc. Dacă este direct pe pagină, poate să conțină elemente diferite care sunt plasate în partea de jos a site-urilor (informații juridice, linkuri către harta site-ului și reguli de utilizare, navigarea repetată etc.).

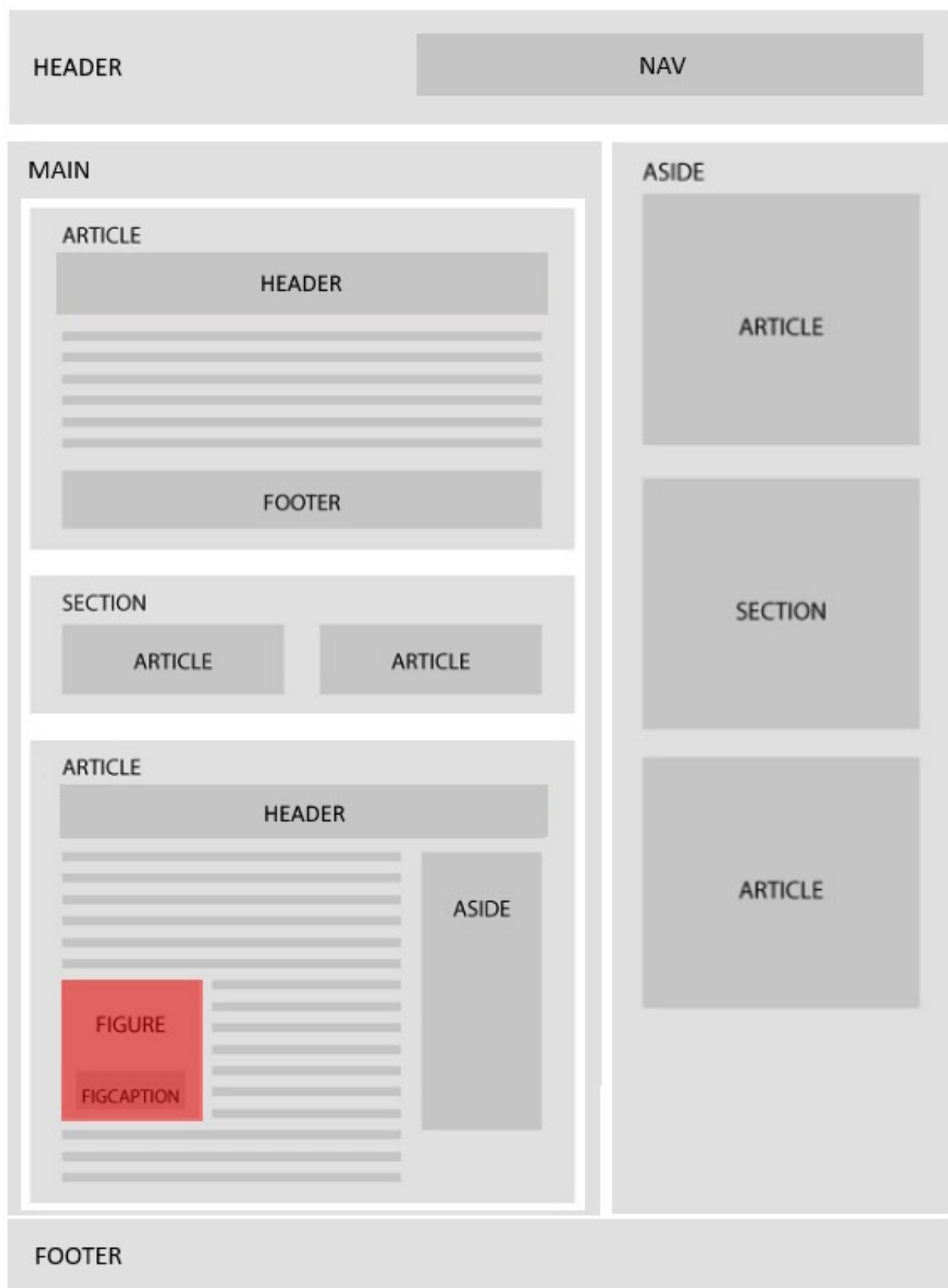


Imaginea 13.10. Exemplu de aspect al unei pagini cu elementele footer

marcate

Figure

Ultimul element este **Figure**. Principalul scop al elementului **<figure>** este de a încercui conținutul suplimentar legat de text (parte din pagină), de exemplu, pentru setarea imaginii, a materialului video, a graficii etc. În el se poate seta și **<figcaption>** lângă conținut, ca și descrierea sa.



Imaginea 13.11. Exemplu de aspect al unei pagini cu elementul figure

marcat

Figcaption este un element opțional și nu trebuie folosit întotdeauna, ci este important doar să rețineți că în cadrul unui element figure poate să apară un figcaption. De exemplu, putem să punem mai multe imagini în elementul figure, dar figcaption rămâne unul pentru toate. Dacă este nevoie ca fiecare imagine să aibă o semnătură specială, postăm și elemente figure și figcaption speciale.

Ca alte elemente semantice HTML5, la fel și figure și figcaption se comportă ca elemente div, respectiv la start au proprietatea display:block;. De asemenea, nu se pune orice stilizare automată pentru aceste elemente, deoarece, așa cum am menționat mai devreme, din punct de vedere vizual nu există nicio diferență între elementele div, header, figure și aside, de exemplu. Acestea au doar importanță semantică, în timp ce pentru stilizare trebuie să avem singuri grijă.

Deseori, întrebarea care se poate auzi în ultima vreme, este: „Când și în ce situație trebuie să folosiți unul sau altul element HTML5 și dacă chiar trebuie folosite sau este mai bine să reveniți la div-ul bine-cunoscut?” Poate că toate acestea vi se par confuze, deoarece unele elemente sunt similare (de exemplu, aside și figure), dar fiecare are scopul său.

În principiu, elementele div se folosesc în continuare, nu sunt eliminate. Putem spune că div se folosește întotdeauna, excepție fiind atunci când există un element definit cu exactitate pentru aplicare (de exemplu, ca header sau footer, și atunci folosim <header>, respectiv <footer>).

Notă:

Să menționăm încă o dată, elementele semantice nu au legătură cu prezentarea vizuală, ele doar oferă informații suplimentare despre o anumită parte din pagină, concret, header spune că va conține informații necesare aflate în partea header-ului, dar el nu este stilizat automat astfel încât să arate ca un header al unui site, adică în

continuare, ca și cu elementele div, aspectul lui trebuie stilizat prin CSS.

Stilizarea elementelor semantice HTML5

Stilizarea CSS a noilor elemente semantice HTML5 se editează la fel ca și la elemente div. Toate elementele, după cum am menționat, la start sunt elemente Block, fără niciun fel de stilizare și, desigur, se supun regulilor modelului box.

Dacă vrem să „tragem” un anumit element cu selectorul CSS, putem face acest lucru în mai multe moduri. Cel mai simplu putem să punem numele (tipul) elementului pentru selector. De exemplu:

```
footer {  
  padding: 10px 25px;  
  background: #f90;  
}
```

dar trebuie să fim atenți, deoarece în acest fel vor fi stilizate toate elementele footer pe pagină, inclusiv footer-ul principal și toate elementele footer laterale, dacă există. Este posibil ca pe site-uri să aveți doar un footer, dar nu este o practică bună să lăsați un astfel de selector.

O practică mai bună este să stilizați un anumit element prin intermediul valorii directe ID sau CLASS sau prin clasa părinte. De exemplu, în HTML putem să avem undeva:

```
<header id="mainHeader">....</header>
```

și ceva mai târziu în pagină:

```
<header class="articleHeader">....</header>
```

Header-ul principal și antetul știrii le-am stiliza, de exemplu, astfel:

```
header#mainHeader {  
  padding:10px 5px;  
  background:#eee;  
}
```

```
header.articleHeader {  
  padding:5px;  
  margin:0 0 10px;  
  border-bottom:1px dashed #ccc;  
}
```

Putem să și lăsăm deoparte cuvântul cheie cu tip tag și să scriem doar **#mainHeader** și **.articleHeader** și autorul trebuie să decidă ce mod va alege, dar poate este util să lăsați numele tag-ului, doar din cauza propriei orientări și pentru o găsimă mai ușoară.

De asemenea, putem să punem și un selector peste elementul părinte. Ne amintim că dacă setăm în selector: elementX + spațiu + elementY, vom nimeri fiecare element Y în cadrul elementului X. În acest caz, putem să scriem, de exemplu:

```
aside#mainSidebar section {...}
```

Dar, și la utilizarea acestui mod trebuie să avem grijă, deoarece, în acest fel vom „ghici” toate elementele section în cadrul elementului aside care are valoarea ID **mainSidebar**.

În materialul multimedia al acestei lecții avem mai multe exemple se

setare și de stilizare a elementelor semantice HTML5.

LINKgroup