

L'objectif de ce projet est de créer un interpréteur pour un langage permettant d'automatiser l'écriture de pages web en HTML/CSS. Dans cette deuxième partie nous vous proposons des exemples qui doivent être interprétés par votre parseur ainsi que le résultat attendu en HTML. Vous pouvez retrouver les différents exemples sur Moodle ainsi que les fichiers sources requis.

Nous rappelons qu'il est plus important d'avoir une bonne structure de données intermédiaire plutôt qu'un résultat en sortie correct.

Pour la sortie vous avez deux possibilités :

- Créer un fichier html (de préférence)
- Écrire le résultat dans le terminal

Enfin nous vous proposons aussi un barème en fin de document.

Exemples

Exemple 1 : page simple

Source

```
!T 'Ma super page'  
!TT 'Quelques paragraphes'  
!P 'Ceci est mon premier paragraphe court'  
!P 'Un paragraphe  
Un peu plus long  
sur trois lignes'  
!TT 'Une image pour le fun'  
!I 'rick.jpg'  
!TTT 'RickRoll'
```

Résultat

```
<!DOCTYPE html>  
<html lang="fr">  
  <head>  
    <meta charset="utf-8">  
    <title></title>  
  </head>  
  <body>  
    <h1>Ma super page</h1>  
    <h2>Quelques paragraphes</h2>  
    <p>Ceci est mon premier paragraphe court</p>  
    <p>Un paragraphe Un peu plus long Sur trois lignes</p>  
    <h2>Une image pour le fun</h2>  
      
    <h3>Rickroll</h3>  
  </body>  
</html>
```

Visuel

Ma super page

Quelques paragraphes

Ceci est mon premier paragraphe court

Un paragraphe Un peu plus long Sur trois lignes

Une image pour le fun



Rickroll

Exemple 2 : page paramétrée

Source

```
@DEFINE (encodage) {'ISO-8859-1'}
%% Utilisation de l'encodage ISO-8859-1 pour faire apparaître des caractères erronés
@DEFINE (icone) {'rick.ico'}
%%%
Une
Jolie
Icône
%%%
@DEFINE (langue) {'fr'}
@TITREPAGE 'rickroll 2'

!T 'Un problème ?'
%% Le caractère è ne doit pas apparaître correctement
```

Résultat

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="iso-8859-1">
    <link rel="icon" type="image/jpg" href="./rickICO.jpg" />
    <title>rickroll 2</title>
  </head>
  <body>
    <!--
      Utilisation de l'encodage ISO-8859-1 pour faire apparaître des caractères erronés
      Une
      Jolie
      Icône
      Le caractère è ne doit pas apparaître correctement
    -->
    <h1>Un problème ?</h1>
  </body>
</html>
```

Visuel



Exemple 3 : page colorée

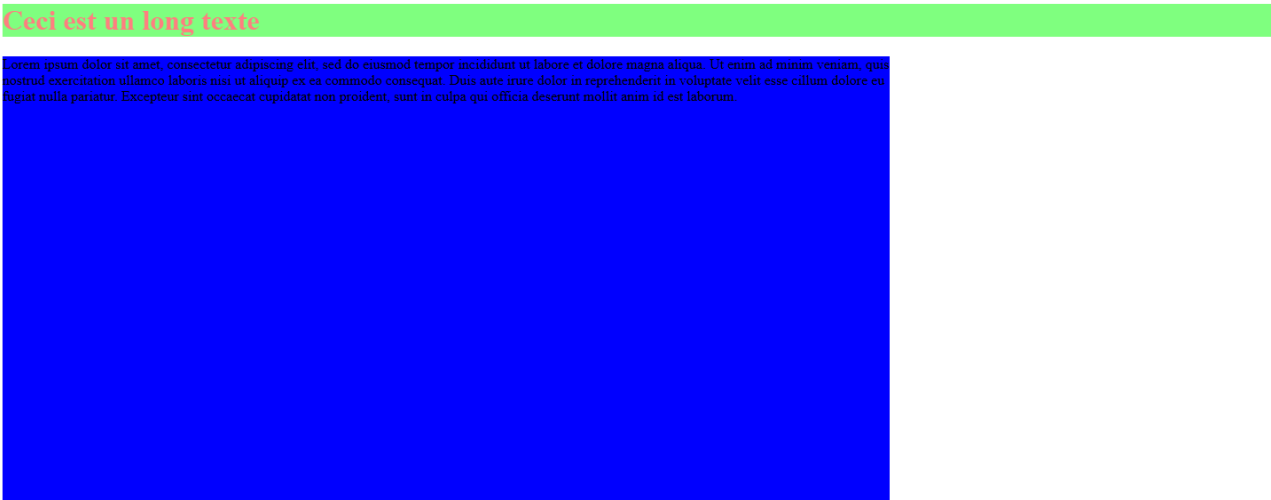
Source

```
!T [couleurTexte : rgb(255,0,0), couleurFond : #00FF00, opacité : 50% ] 'Ceci est un long texte'
!P [largeur : 1000px, hauteur : 500px, couleurFond : #0000ff] 'Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute
irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.'
```

Résultat

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
    <h1 style="color:#ff0000; background-color:#00ff00; opacity:50%;">Ceci est un long texte</h1>
    <p style="width:1000px; height:500px; background-color:#0000ff;">Lorem ipsum dolor sit amet, consectetur adipiscing
  </body>
</html>
```

Visuel



Exemple 4 : Un peu de style

Source

```
@STYLE ( titre1 ) [  
  couleurTexte : rgb(100,255,100)  
  couleurFond : #000000  
]  
  
!T 'Premier Titre'  
!TT 'Premier sous-titre'  
!T 'Deuxième Titre'
```

Résultat

```
<!DOCTYPE html>  
<html lang="fr">  
  <head>  
    <meta charset="utf-8">  
    <title></title>  
  </head>  
  <body>  
    <h1 style="color:#64ff64; background-color:#000000;">Premier Titre</h1>  
    <h2>Premier sous-titre</h2>  
    <h1 style="color:#64ff64; background-color:#000000;">Deuxième Titre</h1>  
  </body>  
</html>
```

Visuel

Premier Titre

Premier sous-titre

Deuxième Titre

Exemple 5 : Les variables et selecteurs

Source

```
indice = 2
couleur1 = #ff0000
couleur2 = rgb(0,255,0)

!P 'Premier Paragraphe'
!P 'Second Paragraphe'
!P 'Troisième Paragraphe'

bloc = !P[indice]
bloc.couleurFond = couleur1

style1 = [couleurFond : couleur2]

bloc = !P[indice-1]
bloc.style = style1
```

Résultat

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
    <p style="background-color:#00ff00;">Premier Paragraphe</p>
    <p style="background-color:#ff0000;">Second Paragraphe</p>
    <p>Troisième Paragraphe</p>
  </body>
</html>
```

Visuel

Premier Paragraphe

Second Paragraphe

Troisième Paragraphe

Exemple 6 : Les conditions

Source

```
!T 'Un titre'

!T[0].couleurFond = #000000

SI (!T[0].couleurFond == #000000) :
!T[0].couleurTexte = #FF0000
FINSI

monTitre = !T[0]

SI (monTitre.couleurTexte == rgb(255,0,0)) :
    monTitre.largeur=500px
SINON:
    monTitre.largeur=800px
FINSI
```

Résultat

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
    <h1 style="background-color:#ff0000; width:500px;">Un titre</h1>
  </body>
</html>
```

Visuel

Un titre

Exemple 7 : Les boucles

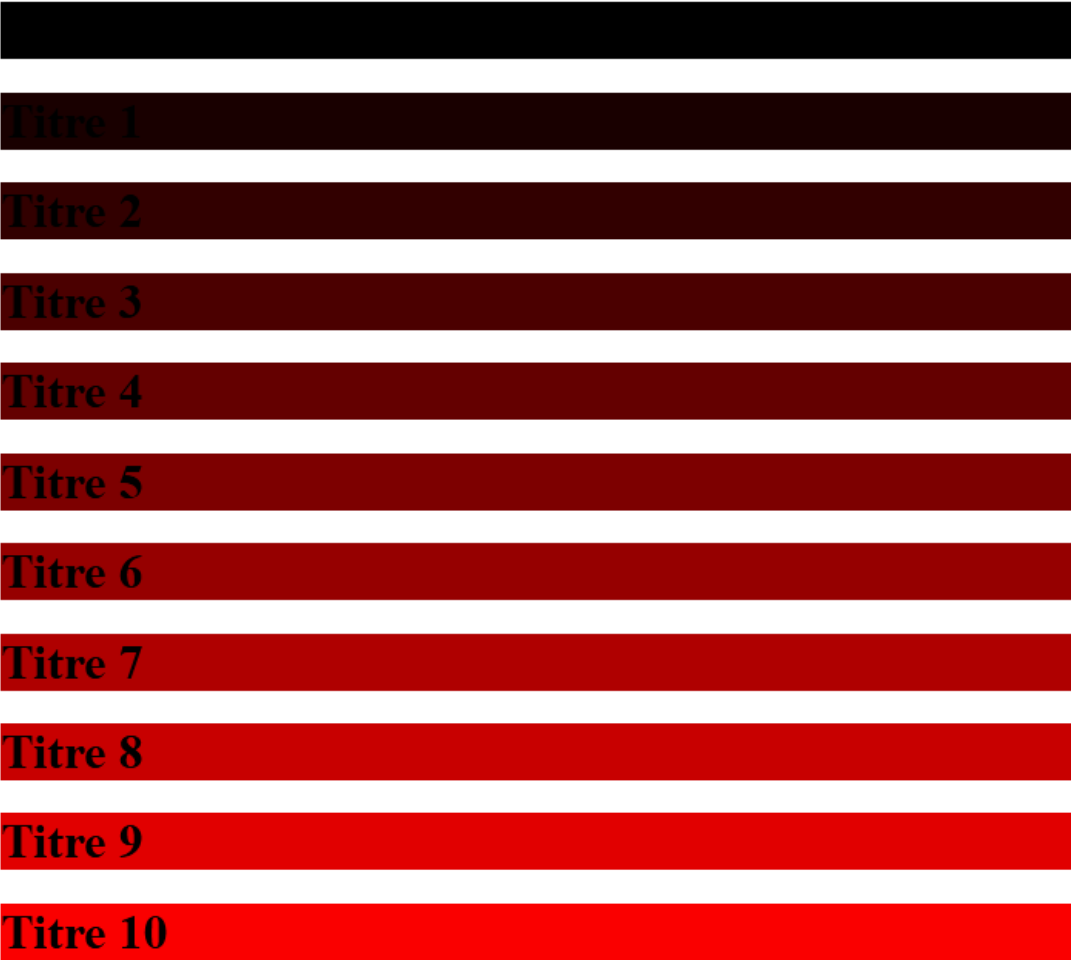
Source

```
POUR i [0,255] +25 :  
  
!T 'Titre ' i/25  
!T[i/25].couleurTexte = rgb(i,0,0)  
FINI
```

Résultat

```
<!DOCTYPE html>  
<html lang="fr">  
  <head>  
    <meta charset="utf-8">  
    <title></title>  
  </head>  
  <body>  
    <h1 style="background-color:#000000;">Titre 0</h1>  
    <h1 style="background-color:#190000;">Titre 1</h1>  
    <h1 style="background-color:#320000;">Titre 2</h1>  
    <h1 style="background-color:#4b0000;">Titre 3</h1>  
    <h1 style="background-color:#640000;">Titre 4</h1>  
    <h1 style="background-color:#7d0000;">Titre 5</h1>  
    <h1 style="background-color:#960000;">Titre 6</h1>  
    <h1 style="background-color:#af0000;">Titre 7</h1>  
    <h1 style="background-color:#c80000;">Titre 8</h1>  
    <h1 style="background-color:#e10000;">Titre 9</h1>  
    <h1 style="background-color:#fa0000;">Titre 10</h1>  
  </body>  
</html>
```

Visuel



Barème indicatif

Afin de vous guider sur la notation du projet voici quelques indications sur le barème.

Il faut d'abord savoir que si seuls l'analyse lexicale et syntaxique sont fonctionnels vous ne pouvez pas avoir plus de la moitié des points, un bon parser doit avoir une structure de données permettant de traiter les différentes parties du programme.

Le barème se base sur les différentes fonctionnalités si votre parser permet de traiter correctement les différents exemples vous pouvez obtenir les points correspondant. Attention si la structure de donnée intermédiaire ne fonctionne pas vous ne pouvez obtenir que la moitié des points au maximum.

- Exemple 1 : 4 points
- Exemple 2 : 2 points
- Exemple 3 : 3 points
- Exemple 4 : 4 points
- Exemple 5 : 3 points
- Exemple 6 : 4 points
- Exemple 7 : 3 points
- Propreté du code : 2 points

Le barème est sur 25 points ramenés à 20 + 2 points bonus sur les éléments suivants :

- Génération d'une sortie correcte : 2 points