



(Cours n° 02)

2^{ème} année Licence Informatique
Université Abou-Bekr Belkaïd - Tlemcen
2018 / 2019

Dr. Mahfoud Houari
mahfoud.houari@gmail.com

Plan du chapitre

1 Introduction

2 Les bases de HTML 5

3 Les tableaux

4 Les formulaires

1 *Introduction*

2 *Création des formulaires*

3 *Validation syntaxique des données d'un formulaire*



Introduction

- Un formulaire est un ensemble de champs permettant de saisir des données.
- Les formulaires sont utilisés pour envoyer des données vers le serveur.
- Les données d'un formulaire sont validées syntaxiquement:
 - ❑ *Automatiquement par le navigateur*
 - ❑ *Grâce aux nouveautés de HTML 5*

Introduction

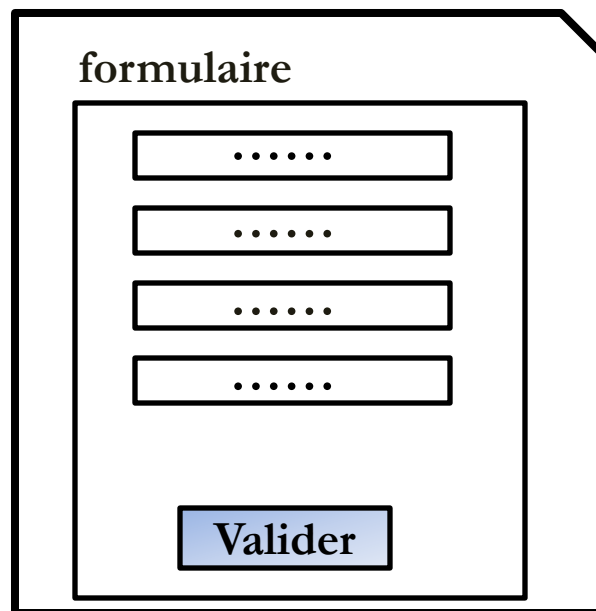
Principe de fonctionnement

Client

Serveur

index.HTML

formulaire



The diagram illustrates a web form structure. It consists of a large outer rectangle representing the HTML document. Inside this rectangle, at the top left, is the text 'formulaire'. Below this text is a smaller rectangle representing the form container. Inside the form container, there are four horizontal input fields, each containing five dots '.....'. Below these input fields is a button labeled 'Valider'.

Introduction

Principe de fonctionnement

Client

Serveur

index.HTML

formulaire

.....

.....

.....

.....

Valider

validation syntaxique

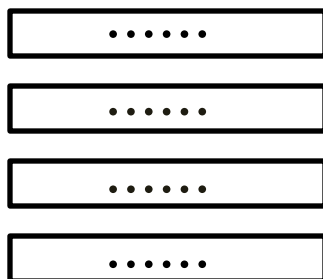
Introduction

Principe de fonctionnement

Client

index.HTML

formulaire



.....

.....

.....

.....

Valider

Serveur

script.PHP

- *Récupération des données*
- *Interaction avec la BDD*
- *Préparation du résultat*

*Transmission
des données*



Introduction

Principe de fonctionnement

Client

index.HTML

formulaire

Valider

*HTML + CSS
+ JavaScript*

Serveur

script.PHP

- *Récupération des données*
- *Interaction avec la BDD*
- *Préparation du résultat*

*Transmission
du résultat*

Introduction

Manipulation "**complète**" d'un formulaire

1. Préciser la méthode de transmission et le script de réception
 2. Créer les champs du formulaire (Balises HTML 5)
 3. Définir les règles de validation syntaxique des données
 4. Programmer le script de réception
-
- The diagram uses red curly braces on the right side of the list to group the tasks. The first three tasks (1, 2, and 3) are grouped under the label *Côté Client*. The fourth task (4) is grouped under the label *Côté Serveur*.
- Côté Client*
- Côté Serveur*

Introduction

Manipulation "**complète**" d'un formulaire

1. Préciser la méthode de transmission et le script de réception

```
<form method="GET ou POST" action="http://127.0.0.1/reception.PHP" >
```

....

```
</form>
```

- ➡ **Attribut **action****: Précise l'adresse du script PHP qui va recevoir et traiter les données du formulaire.
- ➡ **Attribut **method****: Précise la méthode du protocole HTTP qui va être utilisée pour envoyer les données du formulaire.
 - a) **method= "GET"** : données envoyées clairement dans l'URL.
 - b) **method= "POST"** : données envoyées d'une façon transparente.

Plan du chapitre

- 1 Introduction
- 2 Les bases de HTML 5
- 3 Les tableaux
- 4 Les formulaires

- 1 *Introduction*
- 2 *Création des formulaires*
- 3 *Validation syntaxique des données d'un formulaire*



Création des formulaires

Regroupement des champs: éléments *fieldset* et *legend*

Exemple:

```
<form method="POST" action="reception.PHP">  
  <fieldset>  
    <legend>Données personnelles: </legend>  
  </fieldset>  
  <fieldset>  
    <legend>Cursus universitaires: </legend>  
  </fieldset>  
  <fieldset>  
    <legend>Hobbies: </legend>  
  </fieldset>  
</form>
```

Création des formulaires

Regroupement des champs: éléments *fieldset* et *legend*

Exemple: —

Données personnelles: —

Cursus universitaires: —

Hobbies: —

Création des formulaires

Zone de texte: élément *input* avec type égale à *text*

Exemple: —

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Données personnelles:</b></legend>
    <label><b>Nom: </b></label>
    <input type="text" name="nom">
  </fieldset>
</form>
```

Données personnelles: —

Nom:

Création des formulaires

Mot de passe: élément *input* avec type égale à *password*

Exemple: —

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Données personnelles:</b></legend>
    <label><b>Mot de passe: </b></label>
    <input type="password" name="mot_passe">
  </fieldset>
</form>
```

Données personnelles: —

Mot de passe:

Création des formulaires

Adresse URL: élément *input* avec type égale à *url*

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Données personnelles:</b></legend>
    <label><b>Votre URL: </b></label>
    <input type="url" name="adresse_url">
  </fieldset>
</form>
```

Données personnelles:

Votre URL:

Veuillez saisir une URL.

Création des formulaires

Adresse e-mail: élément *input* avec type égale à *email*

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Données personnelles:</b></legend>
    <label><b>Adresse e-mail: </b></label>
    <input type="email" name="adresse_email">
  </fieldset>
</form>
```

Données personnelles:

Adresse e-mail:



Veuillez saisir la partie manquante après le symbole "@".
L'adresse "user@" est incomplète.

Création des formulaires

Date: élément *input* avec type égale à *date*

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Données personnelles:</b></legend>
    <label><b>Date de naissance: </b></label>
    <input type="date" name="date_naissance"
      min="1990-01-01" max="2016-02-18">
  </fieldset>
</form>
```

Données personnelles:

Date de naissance:

Création des formulaires

Date: élément *input* avec type égale à *date*

Exemple:

```
<form method="POST" action="reception.PHP">
```

```
<fieldset>
```

Données personnelles:

Date de naissance:

février 2016 ▾



dim.	lun.	mar.	mer.	jeu.	ven.	sam.
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	1	2	3	4	5

```
</fieldset>  
</form>
```

Création des formulaires

Mois: élément *input* avec type égale à *month*

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Données personnelles:</b></legend>
    <label><b>Mois de naissance: </b></label>
    <input type="month" name="mois_naissance" >
  </fieldset>
</form>
```

Données personnelles:

Mois de naissance: janvier 2016

dim.	lun.	mar.	mer.	jeu.	ven.	sam.
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Création des formulaires

Temps: élément *input* avec type égale à *time*

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Données personnelles:</b></legend>
    <label><b>A quelle heure dormez-vous ?:</b></label>
    <input type="time" name="temps_dodos"
      min="00:00:00" max="03:00:00">
  </fieldset>
</form>
```

Données personnelles:

A quelle heure dormez-vous ?:

Création des formulaires

Nombre: élément *input* avec type égale à *number*

Exemple: —

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Données personnelles:</b></legend>
    <label><b>Votre taille (cm): </b></label>
    <input type="number" name="taille"
      min="150" max="200" step="2">
  </fieldset>
</form>
```

Données personnelles: —

Votre taille (cm):

Création des formulaires

Intervalle: élément *input* avec type égale à *range*

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Contrôle audio:</b></legend>
    <label><b>Ajustez le volume: </b></label>
    <input type="range" name="volume"
      min="0" max="100" step="10">
  </fieldset>
</form>
```

Contrôle audio:

Ajustez le volume:



Création des formulaires

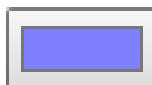
Couleur: élément *input* avec type égale à *color*

Exemple: —

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Données personnelles:</b></legend>
    <label><b>Votre couleur préférée: </b></label>
    <input type="color" name="couleur_pref"
      value="#4169E1" >
  </fieldset>
</form>
```

Données personnelles: —

Votre couleur préférée:



Création des formulaires

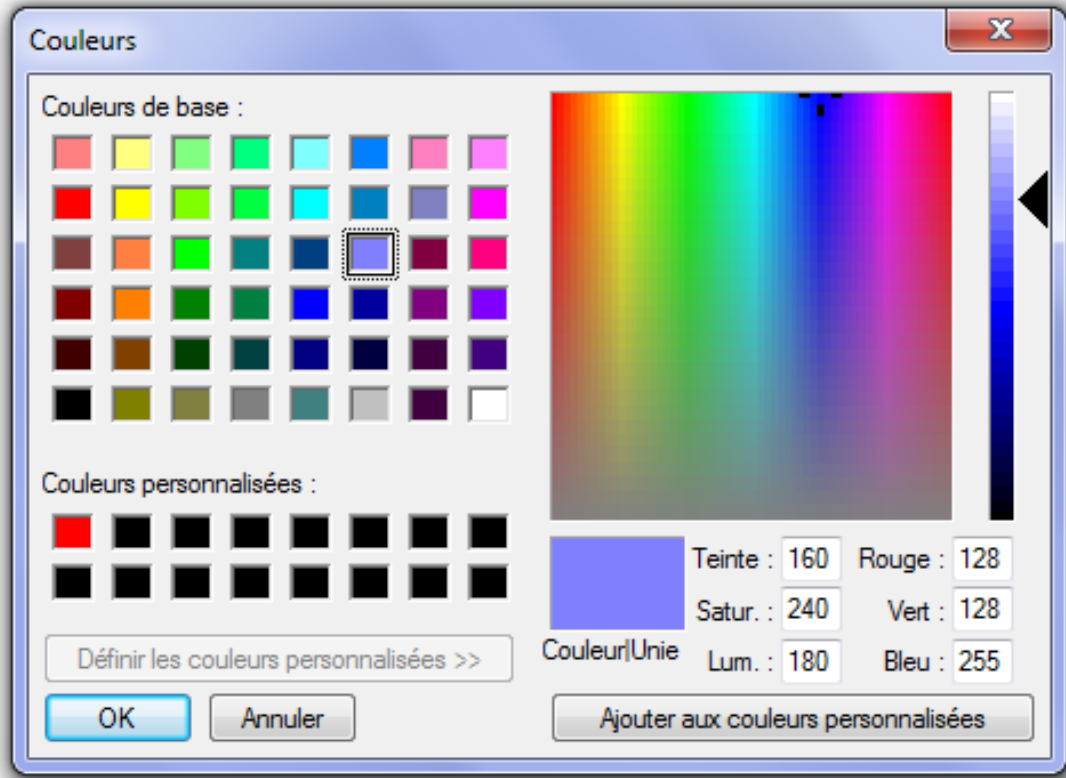
Couleur: élément *input* avec type égale à *color*

Exemple: —

```
<form method:
<fieldset>
  <legend>
    <label>
    <input type="color">
  </fieldset>
</form>
```

Données personnelles: —

Votre couleur préférée:



Création des formulaires

Barre de progression: élément *progress*

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Téléchargement:</b></legend>
    <label><b>Temps restant: </b></label>
    <progress name="progression" value="0.15" >
  </fieldset>
</form>
```

Téléchargement:

Temps restant: 

Création des formulaires

Sélection unique: élément *input* avec type égale à *radio*

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Divers:</b></legend>
    <label><b>Végétarien: </b></label>
    <input type="radio" name="vegetarien" value="oui">(Oui)
    <input type="radio" name="vegetarien" value="non" checked>(Non)
  </fieldset>
</form>
```

Divers:

Végétarien: ☐ (Oui) ☒ (Non)

Création des formulaires

Sélection multiple: élément *input* avec type égale à *checkbox*

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Loisirs:</b></legend>
    <label><b>Football:</b></label>
    <input type="checkbox" name="foot" checked><br>
    <label><b>Natation:</b></label>
    <input type="checkbox" name="natation" checked>
  </fieldset>
</form>
```

Loisirs:

Football: ☒

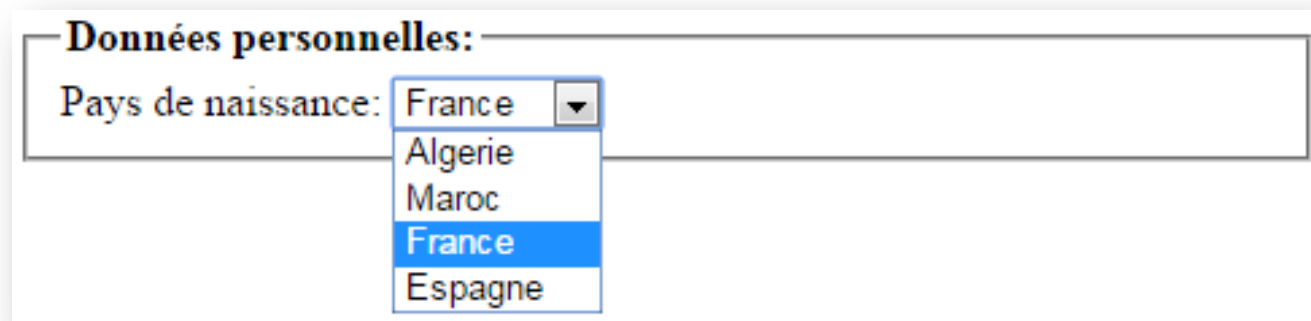
Natation: ☒

Création des formulaires

Liste de choix: éléments *select* et *option*

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Données personnelles:</b></legend>
    <label>Votre ville de naissance:</label>
    <select name="pays_naissance" >
      <option value="algerie">Algerie</option>
      <option value="maroc">Maroc</option>
      <option value="france" selected>France</option>
      <option value="espagne">Espagne</option>
    </select>
  </fieldset>
</form>
```



Données personnelles:

Pays de naissance: France ▼

- Algerie
- Maroc
- France
- Espagne

Création des formulaires

Liste de choix avec catégories: éléments *select* et *option* et *optgroup*

Exemple:

```
<fieldset>
  <legend><b>Données personnelles:</b></legend>
  <label>Pays de naissance:</label>
  <select name="pays_naissance" >
    <optgroup label="Afrique">
      <option value="algerie">Algerie</option>
      <option value="maroc">Maroc</option>
    </optgroup>
    <optgroup label="Europe">
      <option value="france" selected>France</option>
      <option value="espagne">Espagne</option>
    </optgroup>
  </select>
</fieldset>
```

Création des formulaires

Liste de choix avec catégories: éléments *select* et *option* et *optgroup*

Exemple:

```
<fieldset>
```

```
<legend><b>Données personnelles:</b></legend>
```

```
<label>Pays de naissance:</label>
```

```
<select name="pays_naissance" >
```

Données personnelles:

Pays de naissance: France ▼

- Afrique
- Algerie
- Maroc
- Europe
- France
- Espagne

```
</select>
```

```
</fieldset>
```

Création des formulaires

Sélection de fichiers: élément *input* avec type égale à *file*

Exemple: —

```
<form method="POST" action="reception.PHP">  
  <fieldset>  
    <legend><b>Données personnelles:</b></legend>  
    <label>Envoyez votre CV:</label>  
    <input type="file" accept=".PDF, .DOCX">  
  </fieldset>  
</form>
```

Données personnelles: —

Envoyez votre CV: Choisissez un fichier

Aucun fichier choisi

Création des formulaires

Sélection de fichiers: élément *input* avec type égale à *file*

L'attribut **accept** peut avoir les valeurs:

- a) **image/***: pour tous les fichiers image.
- b) **audio/***: pour tous les fichiers audio.
- c) **video/***: pour tous les fichiers video.
- d) **Une extension**: comme **.pdf**, **.png**, **.docx**, **.jpeg**,...
- e) **Plusieurs extensions**: par exemple **accept=".pdf, .docx, image/*"**.

L'attribut booléen **multiple** permet de sélectionner plusieurs fichiers.

Création des formulaires

Texte multi-ligne: élément *textarea*

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Divers:</b></legend>
    <label>Laissez-nous un commentaire:</label><br/>
    <textarea name="commentaire" rows="3" cols="40">
      Votre commentaire nous intéresse...
    </textarea>
  </fieldset>
</form>
```

Divers:

Laissez-nous un commentaire:

Votre commentaire nous intéresse...

Création des formulaires

Réinitialiser ou soumettre les données:

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <legend><b>Action:</b></legend>
    <label>Réinitialisez les données:</label>
    <input type="reset" value="Effacer"><br>
    <label>Envoyez les données:</label>
    <input type="submit" value="Valider"><br>
  </fieldset>
</form>
```

Action:

Réinitialisez les données:

Envoyez les données:

Création des formulaires

Attributs importants:

- a) **value**: valeur par défaut (ou texte des boutons reset et submit).
- b) **size**: la largeur du champs selon le nombre de caractères.
- c) **maxlength**: nombre maximal de caractères de la zone de texte.
- d) **placeholder**: indication du texte à saisir (disparait après le clique).
- e) **required**: attribut booléen qui précise que le champ est obligatoire.
- f) **autocomplete**: avec comme valeur *On* ou *Off*. Cet attribut spécifie si le navigateur peut remplir automatiquement la valeur du champs selon des données préalablement saisies (comme le font les navigateurs avec les champs de mot de passe).
- g) **autofocus**: spécifie que la saisie des données dans le formulaire commence par cet élément.
- h) **pattern**: restriction sur les valeurs à saisir.

Plan du chapitre

1 Introduction

2 Les bases de HTML 5

3 Les tableaux

4 Les formulaires

1 *Introduction*

2 *Création des formulaires*

3 *Validation syntaxique des données d'un formulaire*



Validation syntaxique des données d'un formulaire

- L'attribut **pattern** associe une expression régulière aux champs de texte dont les valeurs saisies doivent respecter cette expression.
- Si l'expression régulière n'est pas respectée alors la validation du formulaire est bloquée (**pas de transmission vers le serveur**)

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <input type="text" pattern="[a-z0-9]">
  </fieldset>
</form>
```



*Un champ de saisie qui accepte uniquement **une seule lettre minuscule** ou **un chiffre***

Validation syntaxique des données d'un formulaire

- L'attribut **pattern** associe une expression régulière aux champs de texte dont les valeurs saisies doivent respecter cette expression.
- Si l'expression régulière n'est pas respectée alors la validation du formulaire est bloquée (**pas de transmission vers le serveur**)

Exemple:

```
<form method="POST" action="reception.PHP">
  <fieldset>
    <input type="text" pattern="[0-9]+">
  </fieldset>
</form>
```

➡ *Un champ de saisie qui accepte un ou plusieurs chiffres (**Donc un nombre**)*

Validation syntaxique des données d'un formulaire

Quelques expressions régulières:

1. [...] : *Intervalle de valeurs permises*
2. [^.....] : *Intervalle de valeurs non permises*
3. [a-z] : *Une seule lettre minuscule*
4. [A-Z] : *Une seule lettre majuscule*
5. [a-z]* : *Zéro ou plusieurs lettres minuscules*
6. [A-Z]+ : *Une ou plusieurs lettres majuscules*
7. [a-z]{8} : *Exactement 8 lettres minuscules*
8. [a-zA-Z]{5, 8} : *Entre 5 et 8 lettres (majuscules ou minuscules)*
9. [a-zA-Z]{5,} : *Au minimum 5 lettres (majuscules ou minuscules)*
10. [^A-Z] : *Aucune lettre majuscule*
11. \d : *Un chiffre quelconque*
12. . : *N'importe quel caractère*
13. \. : *Le point*
14. \- : *Le – (tiret du 6)*
15. (a|b) : *Le **a** ou le **b***
16. (ab) : *Le mot **ab***
17. a(b)? : ***a** suivie d'un **b** optionnel*

Validation syntaxique des données d'un formulaire

Exemple: _____

- Pattern pour un numéro de téléphone fixe :

pattern="043[0-9]{6}"

- Pattern pour un numéro de téléphone mobile :

pattern="0(5 | 6 | 7)[0-9]{8}"

- Pattern pour un pseudo contenant entre 6 et 15 aractères :

pattern="[a-zA-Z][a-zA-Z0-9-_.]{6,15}"

- Pattern pour un mot de passe contenant au moins une lettre en majuscule :

pattern="[a-zA-Z0-9]+[A-Z] [a-zA-Z0-9]+"

Plan du chapitre

- 1 Introduction
- 2 Les bases de HTML 5
- 3 Les tableaux
- 4 Les formulaires
- 5 Divers



Gé-localisation avec Google Maps API

Localiser des établissements:

Exemple:

```
<div align="center" style="background-color:#C4D7ED">
  <h3 align="center">Chercher tous les hôtels d'Oran</h3>
  <iframe
    src="https://maps.google.com/maps?q=hotels%20Oran&z=12&output=embed"
    width="500" height="500" frameborder="0">
  </iframe>
</div>
```

1. L'attribut **q** précise l'endroit que vous cherchez.
2. L'attribut **z** précise le zoom par défaut de la carte (doit être entre 0 et 20).

Gé-localisation



Localisation

Exemple

```
<div align="center">  
  <h3 align="center">Chercher tous les hôtels d'Oran  
  <iframe src="http://maps.google.com/maps?width=100%&height=300px&hl=fr&ie=1" data-bbox="145 140 850 905" style="width:100%; height:300px; border:1px solid black; border-bottom:none; border-top:none; border-left:none; border-right:none; margin:0 auto;"/>  
</div>
```

- 1. L'attribut align="center"
- 2. L'attribut width="100%"

out="embed"

et 20).

Gé-localisation avec Google Maps API

Visualiser un itinéraire:

Exemple:

```
<div align="center" style="background-color:#C4D7ED">
<h3 align="center">Comment nous joindre ?</h3>
<iframe
src="https://maps.google.com/maps?saddr=université%20tlemcen&daddr=aéroport
%20zenata%20tlemcen&z=12&output=embed"
width="500" height="500" frameborder="0">
</iframe>
</div>
```

1. Le paramètre **saddr** précise le lieu de départ (source).
2. Le paramètre **daddr** précise le lieu d'arrivée (destination).

Gé-localisation

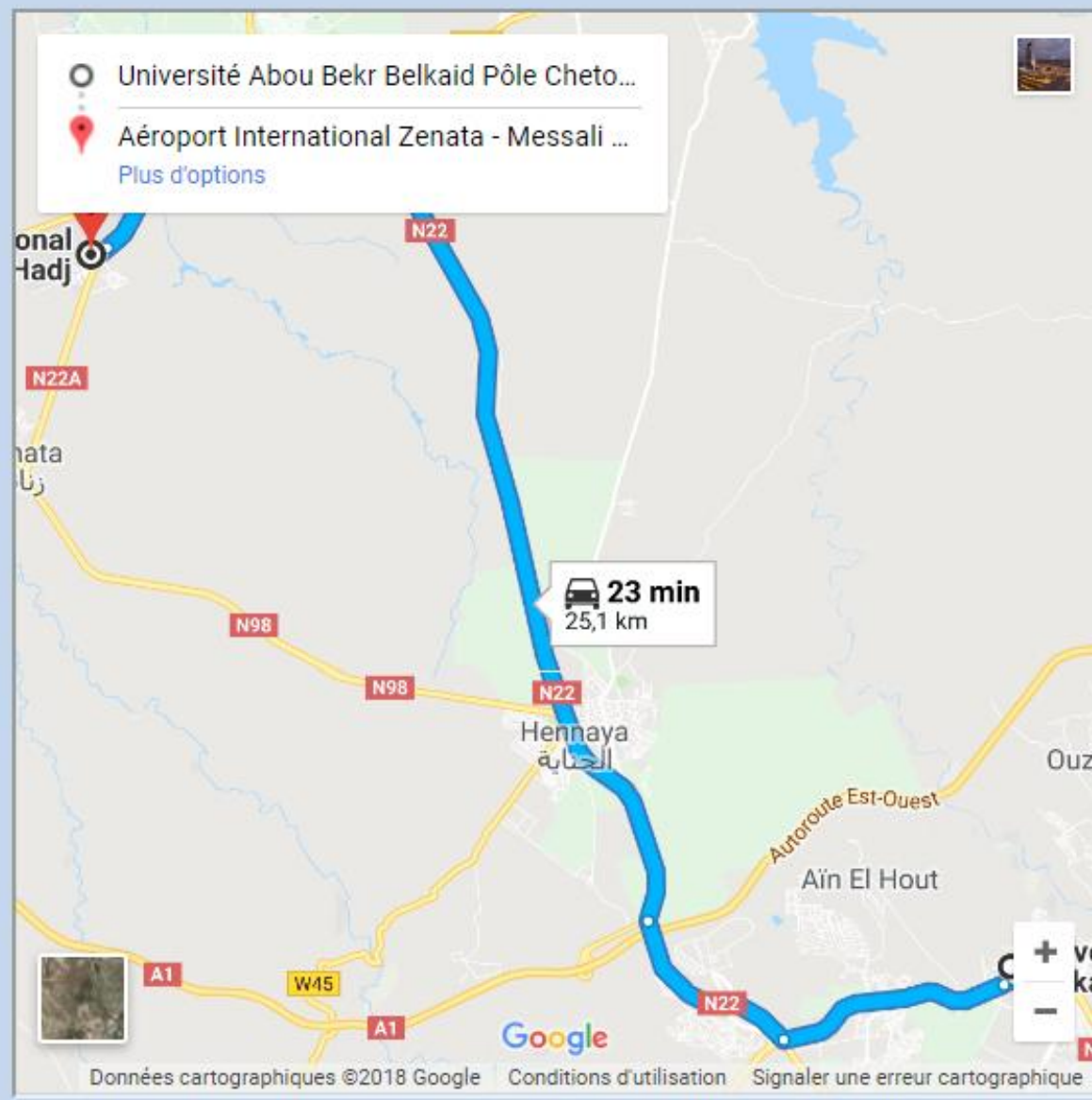
Visualiser un

Exemple:

```
<div align="center">  
<h3 align="center">  
<iframe  
src="https://maps.google.com/maps?addr=aeroporto%20zenata%20messali" width="500" height="300">  
</iframe>  
</div>
```

1. Le paramètre
2. Le paramètre

Comment nous joindre ?



addr=aeroporto