

The background features abstract green geometric shapes. On the left, a solid green triangle points downwards. On the right, a complex arrangement of overlapping, semi-transparent green triangles and polygons creates a layered, architectural effect. The text '08-TD Ajax' is centered in the white space between these two green elements.

# 08-TD Ajax

# Etape 1

- ▶ Reprendre le projet 08-src
- ▶ Rajouter la partie en rouge

Entrer un nombre

## Etape 2

- Après le « submit » Afficher le nombre saisi ici dans la console (utiliser `console.log`)

Entrer un nombre

Submit

## Etape 2 (indication)

- Après le « submit » Afficher le nombre saisi ici dans la console (utiliser console.log)

Entrer un nombre

Submit

```
var tmp=document.getElementById("dyn").value;
```

Confidential - Not for Public Consumption or Distribution

## Etape 3

- Récupérer le nombre saisi et passer ce nombre au fichier « json.php »

Entrer un nombre

- Comment ?

Submit

```
var tmp=document.getElementById("dyn").value;
```

Confidential - Not for Public Consumption or Distribution

## Etape 3

- Récupérer le nombre saisi et passer ce nombre au fichier « json.php »

Entrer un nombre

- Comment ?

```
"json.php?id="
```

```
var tmp=document.getElementById("dyn").value;
```

## Etape 4

► Indication

```
var tmp=document.getElementById("dyn").value;
```

```
var url = "json.php?id=" + tmp;  
xhr.open("GET", url, true);
```

► Vérifier dans « network »

## Etape 5

► Indication

```
var tmp=document.getElementById("dyn").value;
```

```
var url = "json.php?id=" + tmp;  
xhr.open("GET", url, true);
```

- Récupérer le paramètre « id » dans « json.php » et l'afficher avec « echo »
- Comment?



## Etape 5

- ▶ Récupérer le paramètre « id » dans « json.php » et l'afficher avec « echo »
- ▶ Comment?
  - ▶ \$\_GET
- ▶ Vérifier la réponse de l'appel « json.php » dans « network »

## Etape 6

- Filtrer le tableau de retour en renvoyant uniquement l'élément concerné dans le tableau



# Loader

Confidential - Not for Public Consumption or Distribution



# Objectifs

- Mettre un loader au début de l'appel ajax et enlever à la fin



- Comment ?

# Objectifs

- ▶ Mettre un loader au début de l'appel ajax et enlever à la fin



- ▶ Comment ?
  - ▶ Mettre une image à l'appel (balise html)
  - ▶ Supprimer l'image

# Etape 1

- ▶ Ecrire dans la console
  - ▶ « Début appel ajax » au début
  - ▶ « Fin appel ajax » à la fin
- ▶ Amélioration
  - ▶ Rajouter la date et heure en cours

## Etape 2

- ▶ Afficher l'image « loader » au début de l'appel ajax
- ▶ Comment ?

## Etape 2

- ▶ Afficher l'image « loader » au début de l'appel ajax
- ▶ Comment ? Plusieurs solutions possible
  - ▶ Créer l'image dynamiquement (en rajoutant dans le DOM)
  - ▶ L'image existe déjà mais c'est en `display:none`, donc mettre `display:block`



# Accès base de données

Confidential - Not for Public Consumption or Distribution

# Objectifs

- Dans le fichier json.php, remplacer le tableau par un appel vers la base de données Mysql

```
$retour = array(  
    0 => array("Nom"=>"Rakoto", "Prenom"=>"John",  
               "AnneeNaissance"=>1990),  
    1 => array("Nom"=>"Rasoa", "Prenom"=>"Kininina",  
               "AnneeNaissance"=>1994),  
    2 => array("Nom"=>"Rabe", "Prenom"=>"Jean",  
               "AnneeNaissance"=>1993)  
);
```

# Etape 1

- ▶ Créer la base de données
- ▶ Créer la table `Personne(idPersonne, Nom, Prenom, AnneeNaissance)`
- ▶ Insérer des données

## Etape 2

- ▶ Dans json.php
  - ▶ Se connecter à la base
  - ▶ Faire une requête SQL
  - ▶ Retourner un tableau
  - ▶ Mettre dans json\_encode
- ▶ Vérifier la réponse de retour dans « network » , lors de l'appel Ajax