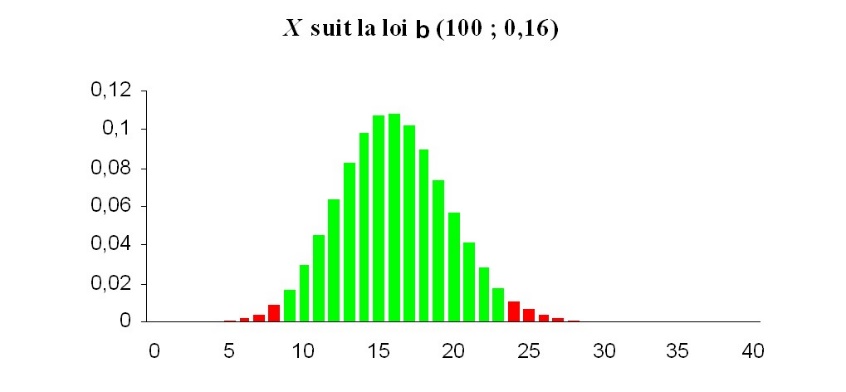
Intervalle de fluctuation à 95%

En PHP



Objectif :

Soit un événement donné dont on connait la probabilité p de réalisation.

On souhaite vérifier sur un échantillon (de taille n) si la fréquence observée associée à un événement se trouve dans un intervalle prévisible à l’aide d’un modèle.

Pour cela tu vas utiliser l’outil informatique pour réaliser un simulateur en PHP.

On souhaite donc pouvoir simuler des échantillons de taille n associé à des tirages d’un dé et effectuer des calculs pour afficher un intervalle de fluctuation. On va donc réaliser une page Web pour cela.

Introduction au langage PHP

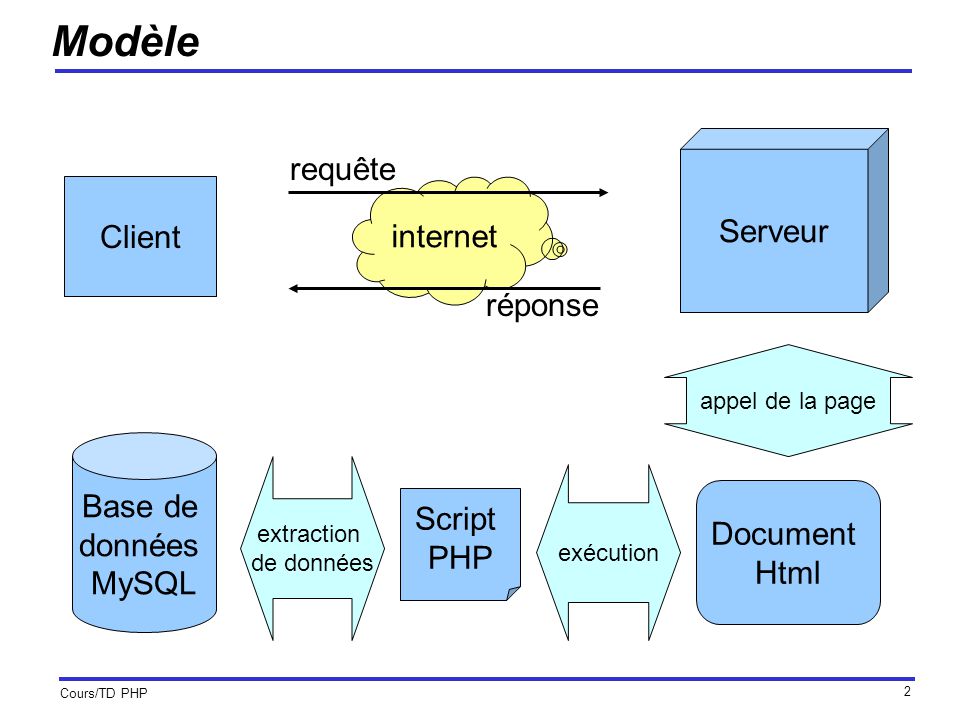
Tu as découvert que pour réaliser une page Web il était possible d’utiliser des balises HTML et des feuilles de styles CSS pour la mise en page.

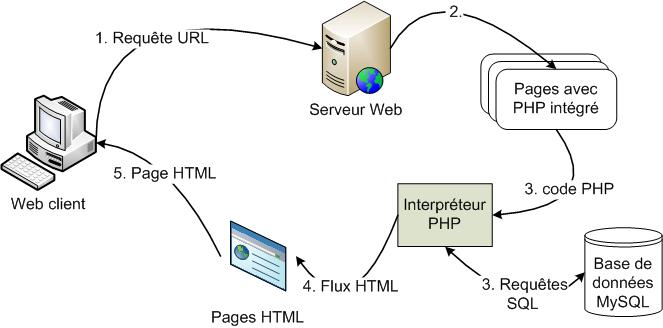
Simplement, les pages Web créées en HTML sont toujours les mêmes et ne peuvent pas être modifiées dynamiquement, comme par exemple afficher la date actuelle sur une page Web.

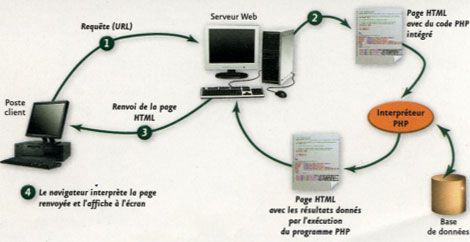
Il faudrait un système qui puisse modifier alors le code HTML.

Le langage PHP est là pour cela. Il va permettre de créer des pages Web dynamiques, c’est-à-dire des pages dont le contenu HTML peut être modifié.

Pour cela il faudra un serveur PHP qui fera la modification du code PHP.

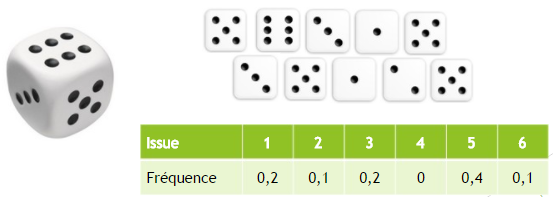
[](http://images.slideplayer.fr/11/3092155/slides/slide_2.jpg)





Les pages PHP que tu vas créer vont pouvoir modifier ta page Web en affichant des résultats non connus à l’avance. Ainsi ta page Web sera dynamique (qui se modifiera en fonction de tes données et des tirages aléatoires).

On souhaite programmer lancement d’un dé à 6 faces. Il y a donc 6 issues différentes possibles.

[](http://images.slideplayer.fr/34/10423545/slides/slide_3.jpg)

L’événement A gagnant est : « le chiffre 2 est gagnant »

Il s’agit d’une expérience de Bernoulli qui consiste à obtenir que deux possibilités, ici, soit on gagne, soit on perd.

Rappelle le calcul de la probabilité P(A).

Simulation d’un lancer de dé

On souhaite afficher sur une page Web des résultats de différents tirages d’un dé à 6 faces.

Tu vas donc réaliser une page Web qui permet de réaliser une telle expérience et afficher la valeur de la face tirée aléatoirement.

Comme en HTML, le PHP utilise des balises et des commandes.

Pour insérer du PHP dans une page HTML on utilise la balise < ?php ….. code php …. ?>

On trouvera alors les codes en PHP entre ces deux balises.

Pour afficher un texte en PHP on utilise la commande echo

Pour effectuer un tirage aléatoire, on peut utiliser la fonction rand(0,32768) qui renvoie un nombre compris dans l’encadrement suivant :

La méthode rand(a,b) renvoie un nombre entier dans l’intervalle suivant :

Pour sauvegarder en mémoire dans une variable, on utilise un nom de variable. En PHP, une variable commence toujours par $

$nb\_aleatoire=rand(0,32768) ; par exemple.

Pour afficher un texte suivi d’une variable on écrit la commande :

Echo « mon texte ». $ma\_variable ;

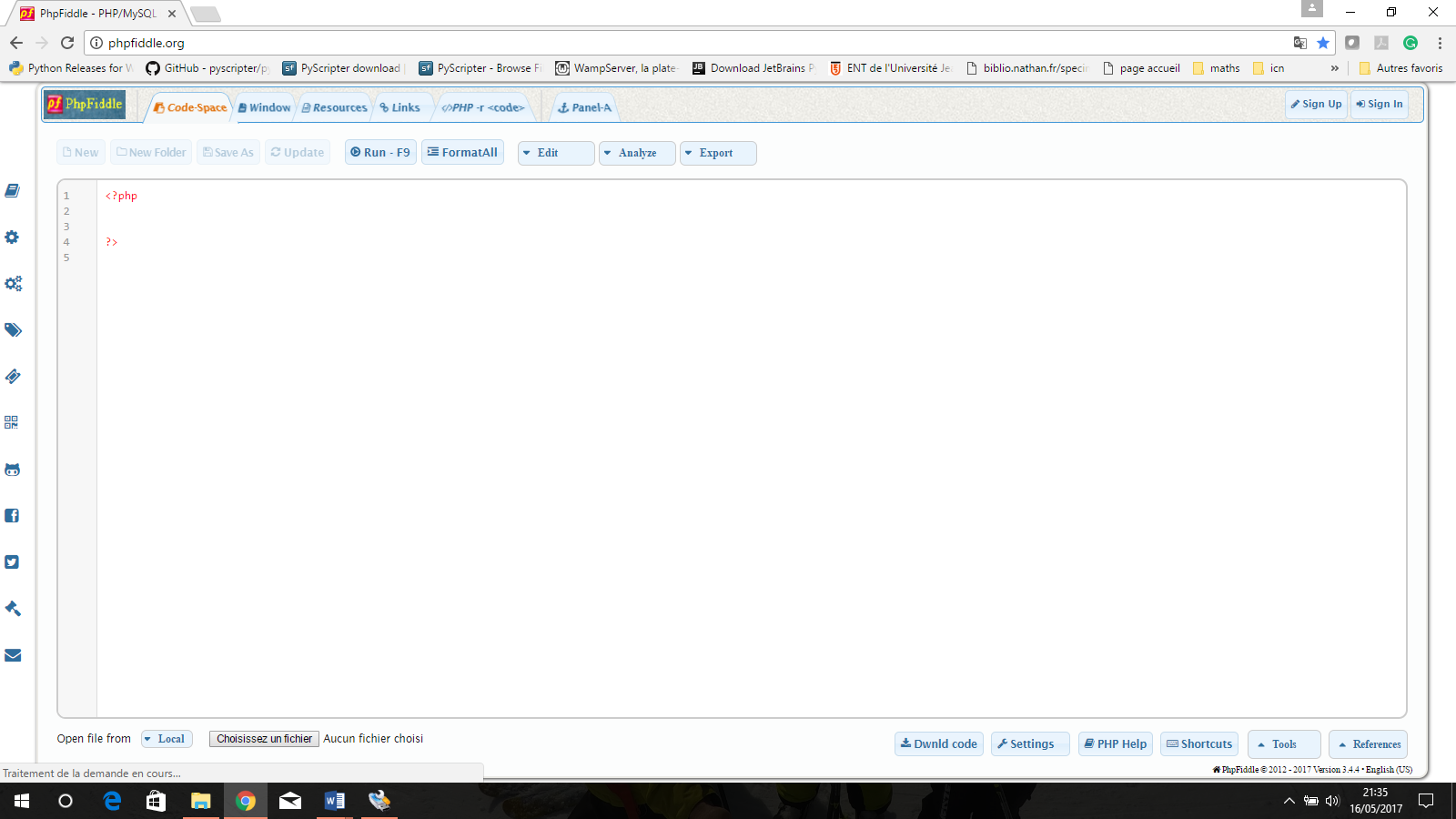
Attention n’oublie pas d’ajouter un point-virgule à la fin de chaque commande PHP !

Réalise alors ta page en PHP. Pour cela tu as vu que pour que la page web se fabrique à partir d’un code PHP qui a besoin d’un serveur PHP.

Tu vas utiliser pour cela un site qui héberge déjà un serveur PHP.

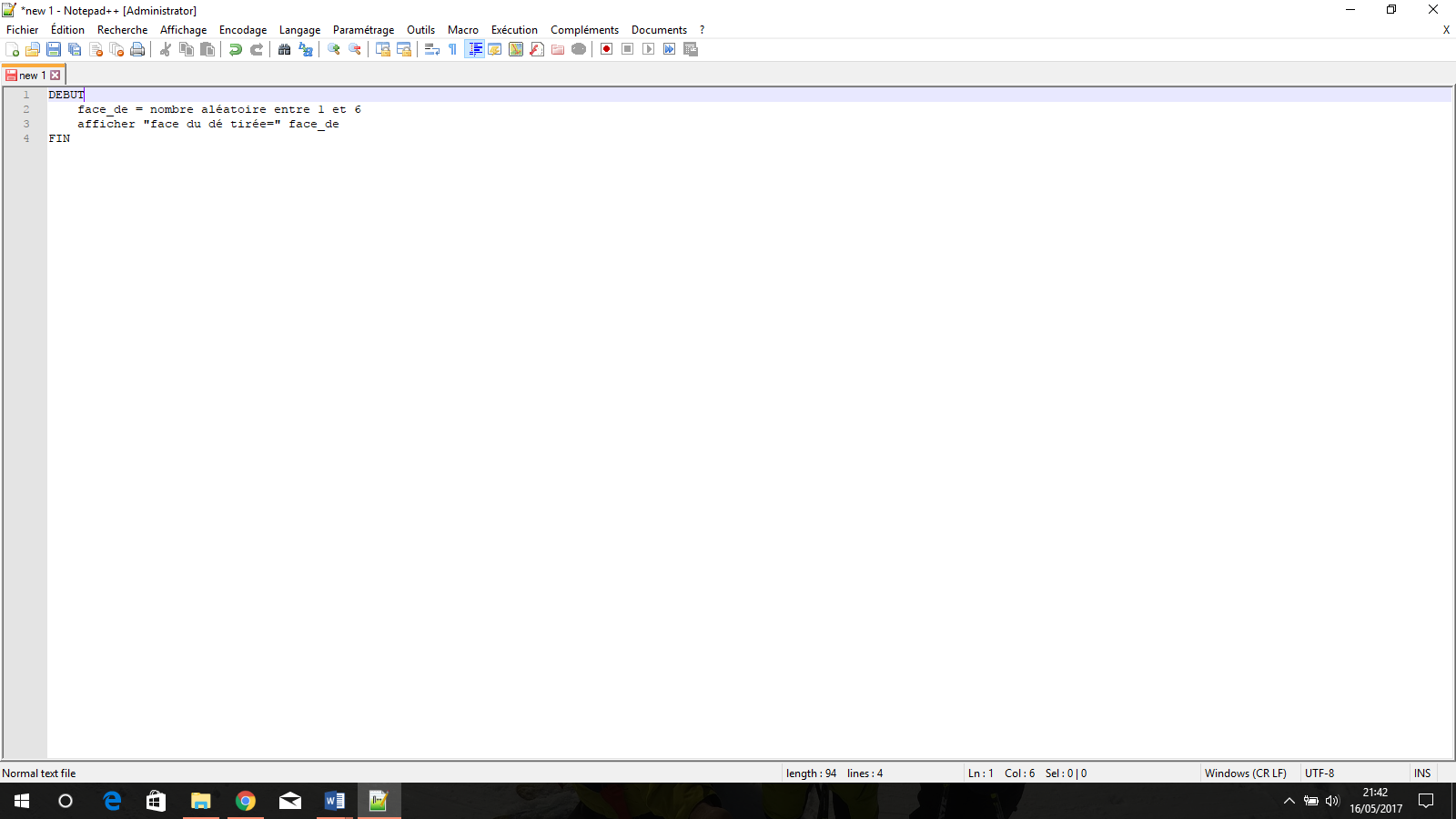
Connecte-toi sur : <http://phpfiddle.org/>

Tu as alors à ta disposition une page en PHP dans l’onglet « code\_space »



Tape alors ton code pour afficher une valeur comprise entre 0 et 6 qui va simuler une valeur aléatoire qui va simuler le lancer d’un dé à 6 faces.

Algorithme :

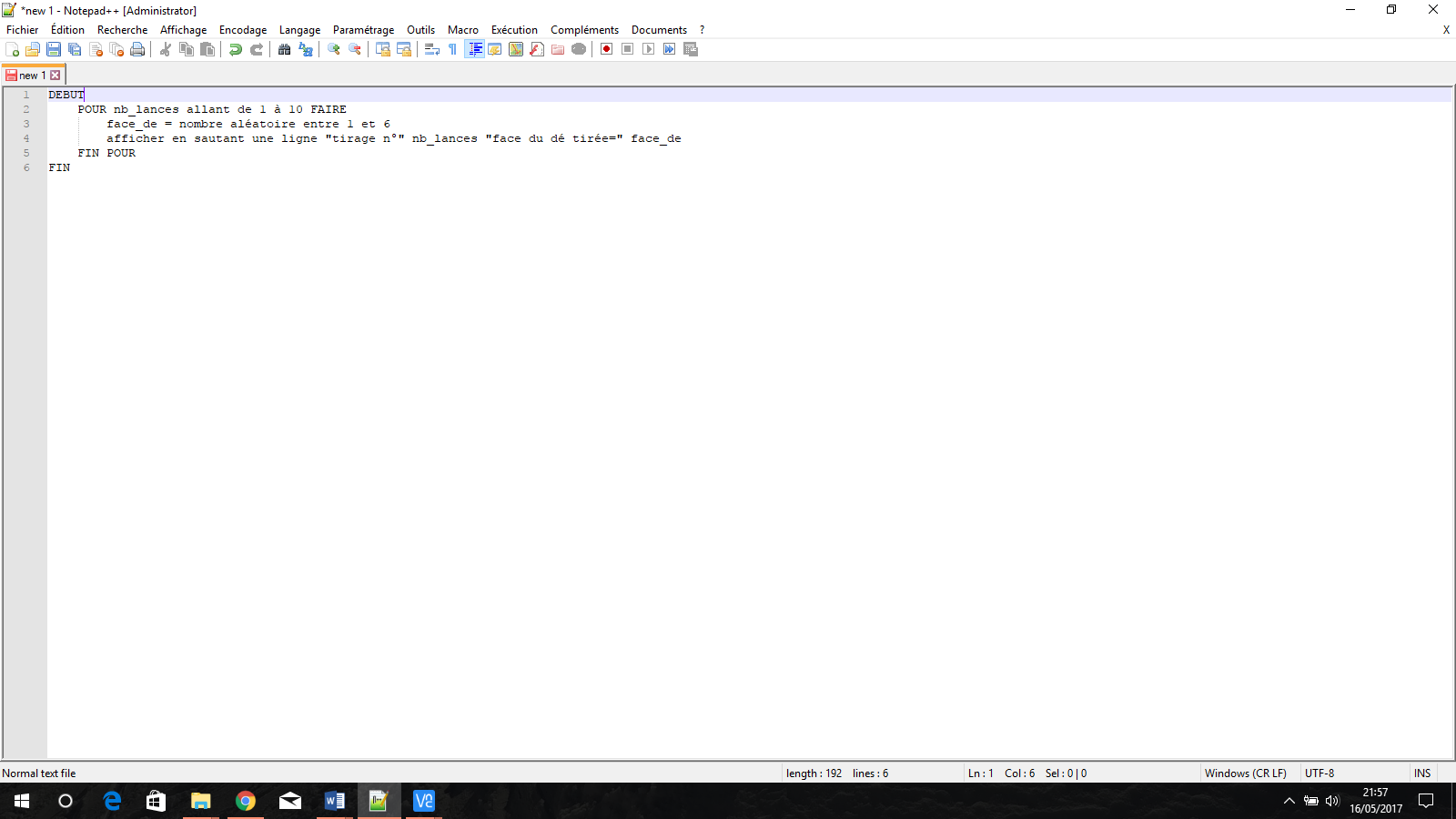


Pour tester ton code PHP, clique sur RUN 

Simulation de 10 lancers de dé

Réalise alors un programme qui affiche 10 tirages (=1 échantillon de taille 10) successifs aléatoires.

Algorithme :



Attention, tu devras ajouter en fin de commande echo une balise HTML pour ajouter un saut de ligne. Rappelle-toi que le PHP génère au final un page WEB !

Pour réaliser une boucle bornée en PHP on utilise

For($i=valeur\_départ ;$i<=valeur\_fin ;$i++){

On tape son code

}

On affiche ainsi un échantillon de taille n=10.

Un échantillon est un sous-ensemble obtenu par prélèvement aléatoire dans une population.

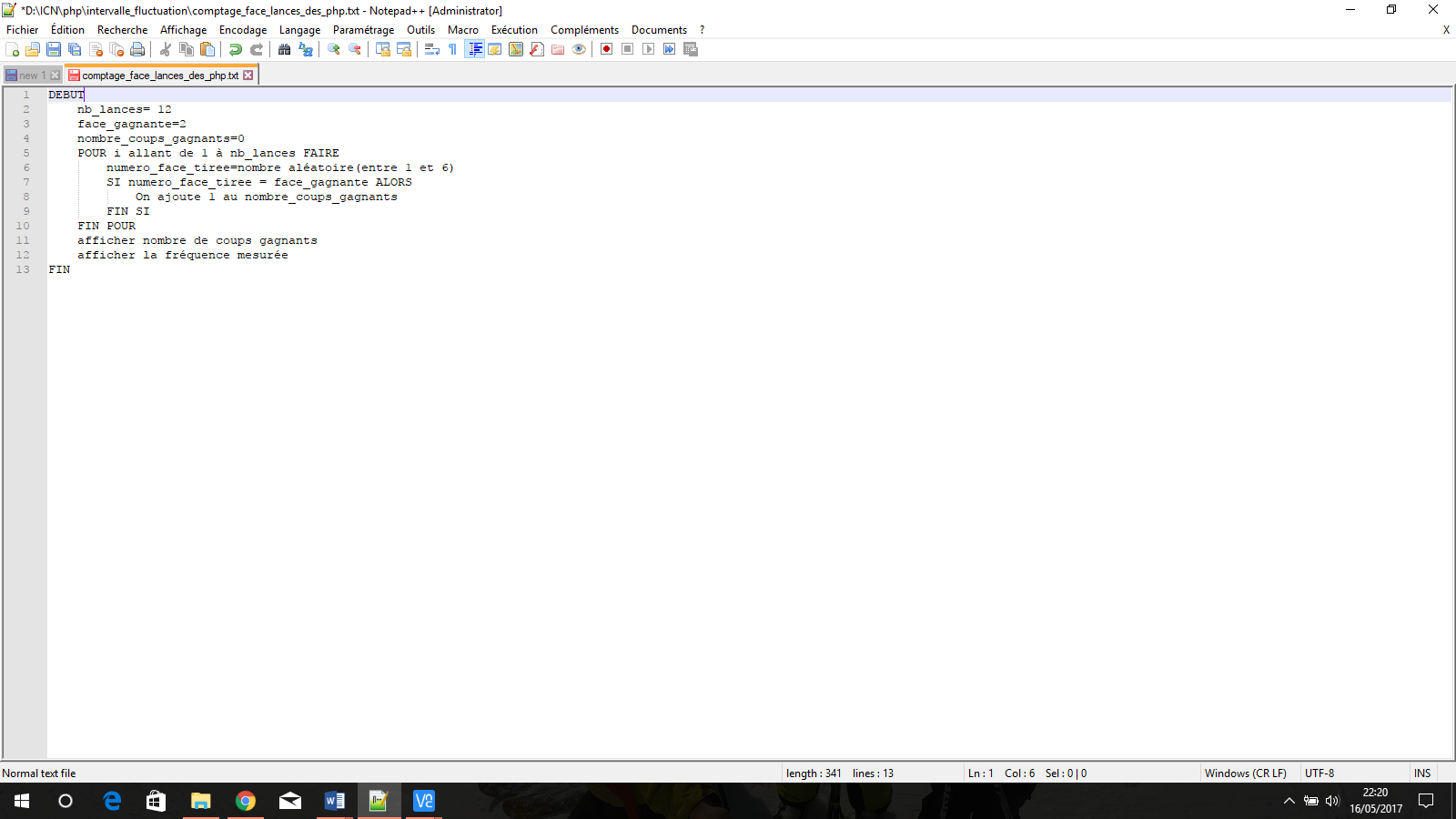
Répétition n fois d’une même expérience (1 échantillon de taille n)

On va réaliser la même expérience que précédemment, mais cette fois-ci, on va compter le nombre de fois où se réalise l’élément gagnant et mémorisant le résultat dans une variable X (appelée variable aléatoire)



On ajoutera une variable en début de programme le nombre de tirages successifs à faire (ce sera n = taille de l’échantillon). On utilisera pour cela $nb\_lances=12 ; par exemple

Algorithme :

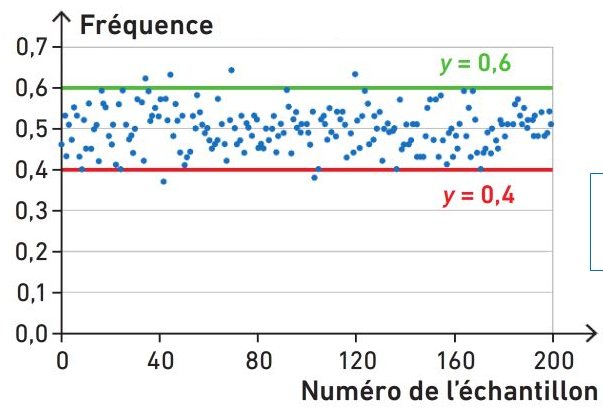


Complète alors le tableau suivant

|  |  |  |  |
| --- | --- | --- | --- |
| Taille échantillon (**n=10**) | Echantillon n°1 | Echantillon n°2 | Echantillon n°3 |
| Fréquence observée (f) |  |  |  |
| Taille échantillon (**n=100**) | Echantillon n°1 | Echantillon n°2 | Echantillon n°3 |
| Fréquence observée (f) |  |  |  |
| Taille échantillon (**n=1000**) | Echantillon n°1 | Echantillon n°2 | Echantillon n°3 |
| Fréquence observée (f) |  |  |  |

Que constates-tu ?

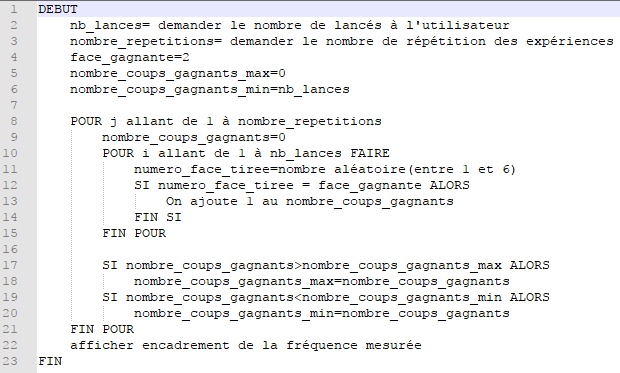
Répétition de la mesure de la fréquence (plusieurs échantillons de taille n)

Si tu relances plusieurs fois le programme précédent est-ce que la fréquence mesurée est toujours la même pour n=30, n étant le nombre de fois où on répète la même expérience de Bernoulli ?

Tu vas alors mesurer quelle est la fréquence mesurée la plus faible (min) et la plus grande(max) parmi toutes les fréquences mesurées.

Au lieu de noter toutes ces fréquences successivement, ce qui serait un travail fastidieux, tu vas modifier à nouveau ton programme pour te permettre de trouver ces deux valeurs de freqmin et freqmax.

Algorithme :



Pour réaliser un test de condition SI en PHP

If ($variable1 < $variable2) {

Mettre le code associé au test vrai

}

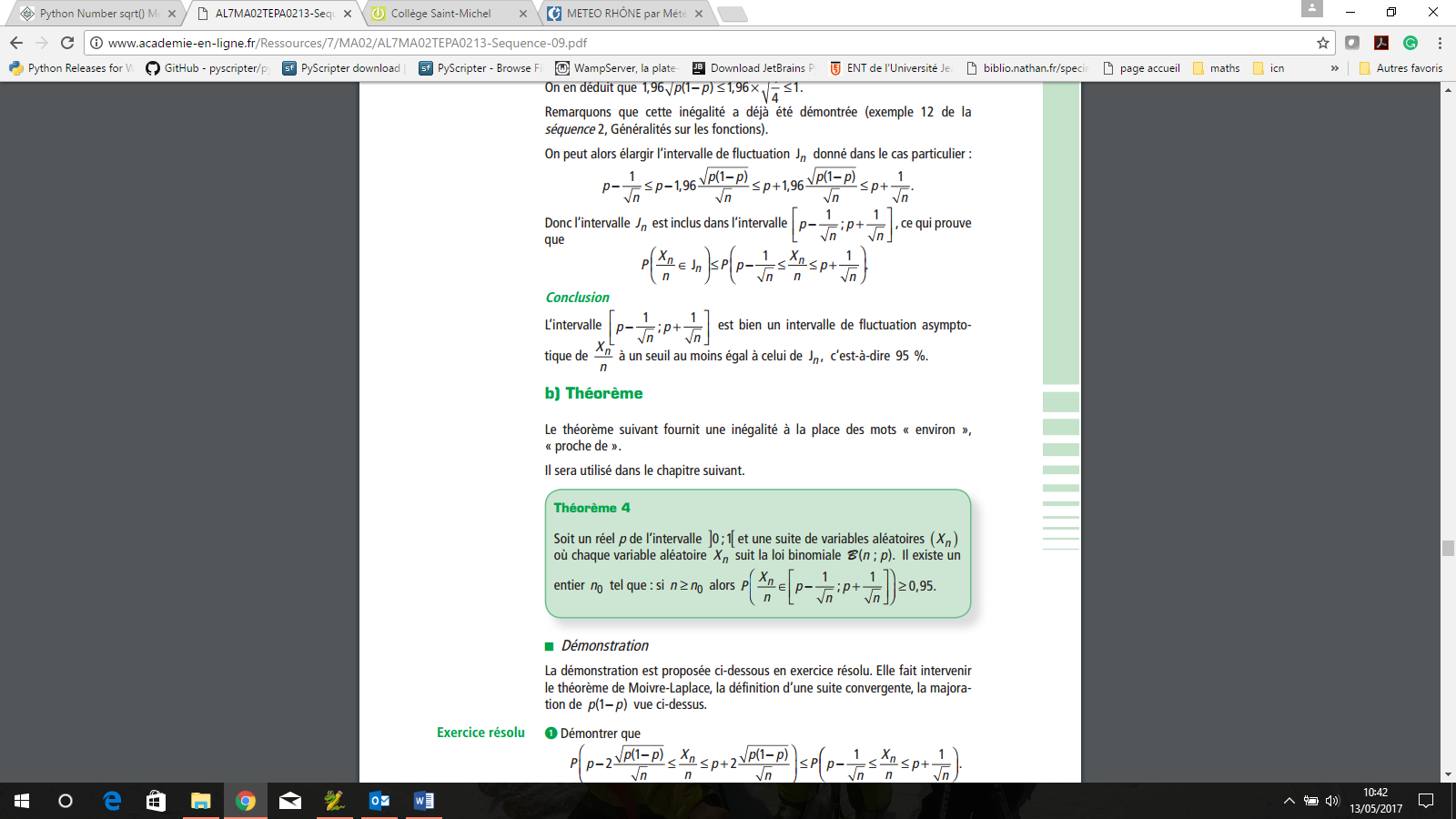
Tu observes un minimum et un maximum pour la fréquence mesurée qui évolue dans un intervalle de fluctuation avec une probabilité de 0,95.

Donne cet intervalle pour n=30 (nombre lancés de dé) et un nombre de répétition de 100.

Note cet intervalle de fluctuation de la fréquence.

Comparaison avec l’intervalle de fluctuation

Tu as vu en cours de mathématiques qu’il est possible de prévoir un encadrement la fréquence avec une probabilité de 0,95.



Modifie alors ton programme pour afficher l’intervalle de fluctuation estimé mathématiquement avec une probabilité de 0,95

Tu auras besoin de la méthode racine carrée (sqrt).

Puis pour trouver la racine carré d’un nombre tape sqrt(ton\_nombre).

Compare alors ton encadrement à celui de la prévision mathématique. Est-ce que le prévision est cohérente ?

Amélioration avec Interface

Tu vas améliorer ta page web pour que lorsque tu te connectes sur son serveur Web (ta page php de départ en fait). Ta page te demandera alors le nombre de répétitions d’échantillons ainsi que la taille n de l’échantillon.

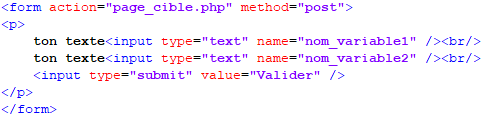
Il te faudra créer un compte (sign up).

Pour cela on va utiliser un mécanisme particulier. On va utiliser 2 pages php.

La première page php de départ sera la page dans laquelle tu vas demander les valeurs à l’aide d’un formulaire.

Puis ces valeurs seront ensuite passées à une autre pages PHP qui pourra alors les lire comme des variables et réaliser l’algorithme précédent avec peu de modifications.

Conception de la première page (formulaire HTML)

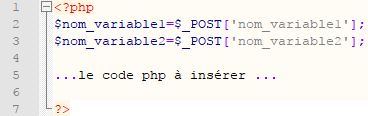


Sauvegarde ta page PHP 





Conception de la page\_cible en PHP

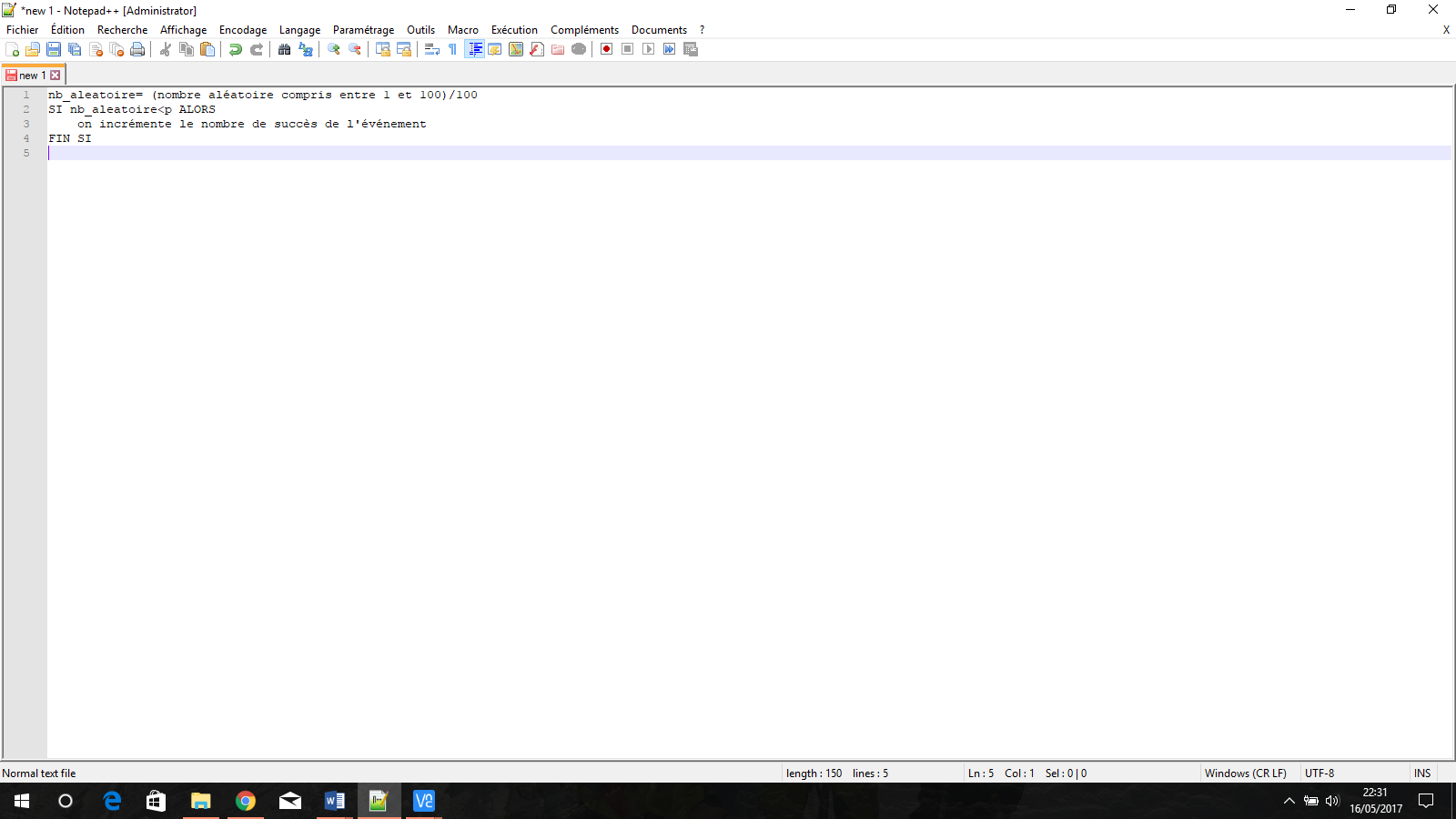


Pour tester, ouvre ta page de départ, puis RUN.

Saisie de la probabilité p associée à un événement A donné

Modifie à nouveau ton code PHP pour saisir la probabilité p que tu souhaites et qui correspond à des événements différents.

Il te faudra modifier alors une partie de ton algorithme. En effet, on ne parle plus de face de dé, mais de probabilité p=P(A) d’un événement A. Il faudra alors tirer un nombre aléatoire décimal compris en 0 et 1. Car une probabilité est bien comprise entre 0 et 1.



Synthèse

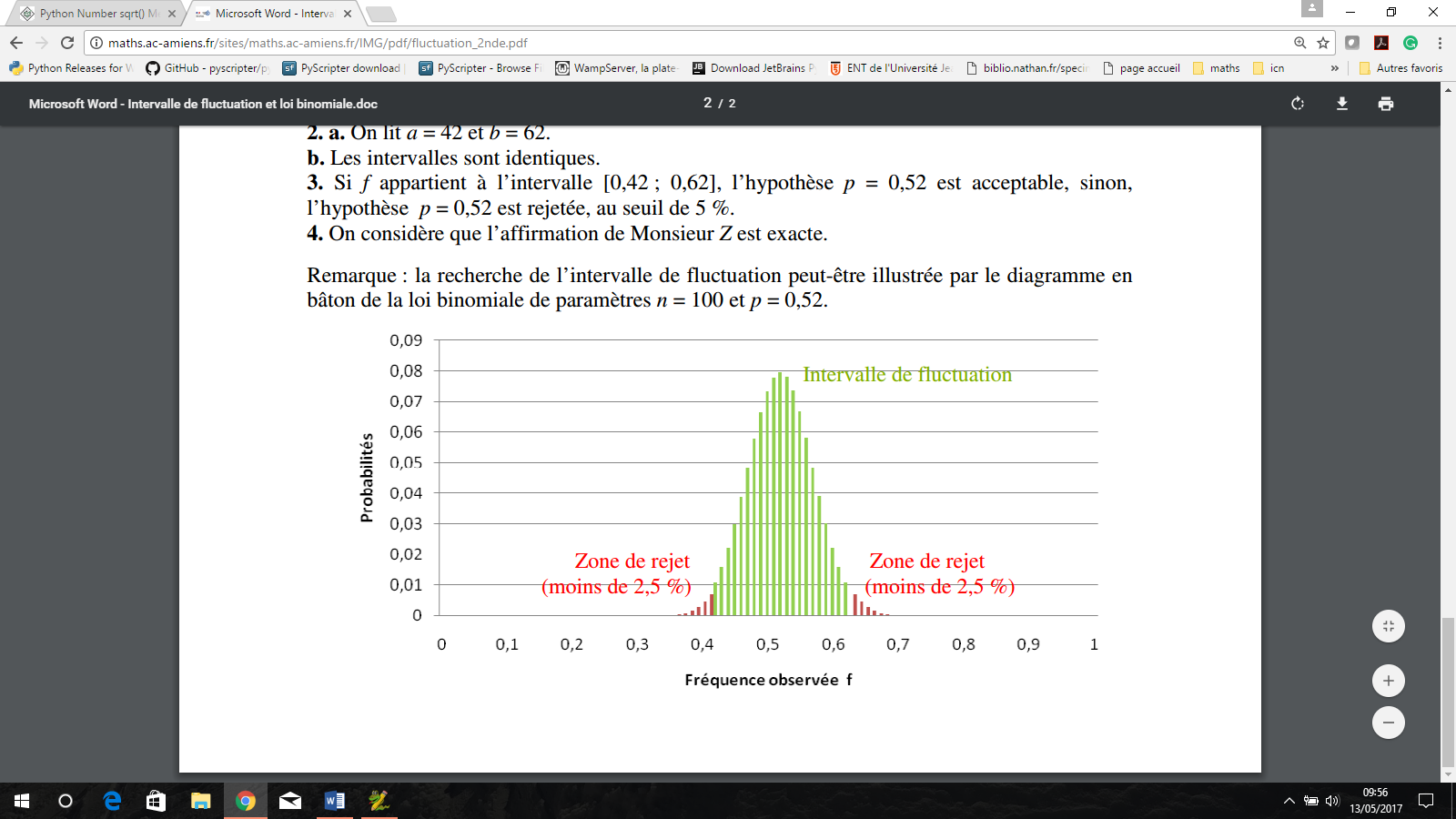
La fréquence f des individus possédant le caractère dans l’échantillon varie d’un échantillon à l’autre : c’est la fluctuation d’échantillonnage

En fonction de l’appartenance ou non de f à l’intervalle de fluctuation à 0,95 que l’on a déterminé, on prend une décision concernant la conformité de l’échantillon :

si f n’appartient pas à l’intervalle, on rejette, au risque d’erreur de 5 %, l’hypothèse que l’échantillon est compatible avec le modèle

dans le cas contraire, on ne peut pas rejeter l’hypothèse

L’intervalle de fluctuation à 95 % d’une fréquence correspondant à la réalisation, sur un échantillon aléatoire, d’une variable aléatoire X de loi B(n ; p) est l’intervalle [a/n ;b/n]

[](http://maths.ac-amiens.fr/sites/maths.ac-amiens.fr/IMG/pdf/fluctuation_2nde.pdf)

a est le plus petit entier tel que P(X ≤ a) > 0, 025 ; b est le plus petit entier tel que P(X ≤ b) > 0, 975.

Exercices d’application

Modifier le programme pour répondre à cette question

On dispose d’un dé bien équilibré, on gagne quand on obtient 1 ou 6. Déterminer un intervalle de fluctuation au seuil de 95 % de la fréquence des lancers gagnants dans les échantillons de taille 100.

Corréler la réponse des par votre ordinateur à celle de la théorie (faites vos calculs). Retrouves-tu les mêmes informations ? Conclure.

Soit l’expérience de Bernouilli suivante :

Prend une pièce de monnaie réelle. Effectue 30 lancés.

Tu vas vérifier si ta pièce est bien équilibrée. Pour cela tu vas comptabiliser le nombre de côtés face.

Vérifie à l’aide de ton application si l’hypothèse : « ma pièce est bien équilibrée » est juste.

On rappelle la règle de décision :

Règle de décision : si f appartient à l’intervalle I , n on décide que la pièce est équilibrée ; si f n’appartient pas à l’intervalle I n , on décide que la pièce n’est pas équilibrée.

Vérifier l’intervalle de fluctuation à 95%. Pour cela compter le nombre de fréquences observées (simulées) qui se trouve en dehors de l’intervalle de fluctuation prévisible et vérifier la probabilité :

