



Extrait du Collège et Lycée Camille Saint-Saëns

<http://stsaens-lyc.spip.ac-rouen.fr/spip.php?article68>

# PS2 pour le 07 Septembre 2009

- Les activités en classe - Sciences - Sciences de la vie et de la Terre - Exercices -



Date de mise en ligne : dimanche 6 septembre 2009

## **Description :**

Un petit exercice suite à la pré-rentree.

---

**Copyright © Collège et Lycée Camille Saint-Saëns - Tous droits réservés**

---

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "xhtml11.dtd">
```

```
Signification de l'expression Grippe A H1 N1 /* This is the CSS stylesheet used in the exercise. */ /* Elements in
square brackets are replaced by data based on configuration settings when the exercise is built. */ /*
BeginCorePageCSS */ /* Made with executable version Java Hot Potatoes 6.0.2.8 */ /* Hack to hide a nested
Quicktime player from IE, which can't handle it. */ /* html object.MediaPlayerNotForIE { display: none; } body{
font-family: Geneva,Arial; background-color: #FFFFFF; color: #000066; margin-right: 5%; margin-left: 5%;
font-size: small; } p{ text-align: left; margin: 0px; font-size: 100%; } table,div,span,td{ font-size: 100%; color:
#000066; } div.Titles{ padding: 0.5em;; text-align: center; color: #000066; } button{ font-family: Geneva,Arial;
font-size: 100%; display: inline; } .ExerciseTitle{ font-size: 140%; color: #000066; } .ExerciseSubtitle{ font-size:
120%; color: #000066; } div.StdDiv{ background-color: #DDDDFF; text-align: center; font-size: 100%; color:
#000066; padding: 0.5em; border-style: solid; border-width: 1px 1px 1px 1px; border-color: #000066;
margin-bottom: 1px; } /* EndCorePageCSS */ .RTLText{ text-align: right; font-size: 150%; direction: rtl;
font-family: "Simplified Arabic", "Traditional Arabic", "Times New Roman", Geneva,Arial; } .CentredRTLText{
text-align: center; font-size: 150%; direction: rtl; font-family: "Simplified Arabic", "Traditional Arabic", "Times New
Roman", Geneva,Arial; } button p.RTLText{ text-align: center; } .RTLGapBox{ text-align: right; font-size: 150%;
direction: rtl; font-family: "Times New Roman", Geneva,Arial; } .Guess{ font-weight: bold; } .CorrectAnswer{
font-weight: bold; } div#Timer{ padding: 0.25em; margin-left: auto; margin-right: auto; text-align: center; color:
#000066; } span#TimerText{ padding: 0.25em; border-width: 1px; border-style: solid; font-weight: bold; display:
none; color: #000066; } span.Instructions{ } div.ExerciseText{ } .FeedbackText, .FeedbackText
span.CorrectAnswer, .FeedbackText span.Guess, .FeedbackText span.Answer{ color: #000066; } .LeftItem{
font-size: 100%; color: #000066; text-align: left; } .RightItem{ font-weight: bold; font-size: 100%; color: #000066; }
span.CorrectMark{ } input, textarea{ font-family: Geneva,Arial; font-size: 120%; } select{ font-size: 100%; }
div.Feedback { background-color: #FFFFFF; left: 33%; width: 34%; top: 33%; z-index: 1; border-style: solid;
border-width: 1px; padding: 5px; text-align: center; color: #000066; position: absolute; display: none; font-size:
100%; } div.ExerciseDiv{ color: #000066; } /* JMatch flashcard styles */ table.FlashcardTable{ background-color:
transparent; color: #000066; border-color: #000066; margin-left: 5%; margin-right: 5%; margin-top: 2em;
margin-bottom: 2em; width: 90%; position: relative; text-align: center; padding: 0px; } table.FlashcardTable tr{
border-style: none; margin: 0px; padding: 0px; background-color: #DDDDFF; } table.FlashcardTable td.Showing{
font-size: 140%; text-align: center; width: 50%; display: table-cell; padding: 2em; margin: 0px; border-style: solid;
border-width: 1px; color: #000066; background-color: #DDDDFF; } table.FlashcardTable td.Hidden{ display: none;
} /* JMix styles */ div#SegmentDiv{ margin-top: 2em; margin-bottom: 2em; text-align: center; } a.ExSegment{
font-size: 120%; font-weight: bold; text-decoration: none; color: #000066; } span.RemainingWordList{ font-style:
italic; } div.DropLine { position: absolute; text-align: center; border-bottom-style: solid; border-bottom-width: 1px;
border-bottom-color: #000066; width: 80%; } /* JCloze styles */ .ClozeWordList{ text-align: center; font-weight:
bold; } div.ClozeBody{ text-align: left; margin-top: 2em; margin-bottom: 2em; line-height: 2.0 } span.GapSpan{
font-weight: bold; } /* JCross styles */ table.CrosswordGrid{ margin: auto auto 1em auto; border-collapse:
collapse; padding: 0px; background-color: #000000; } table.CrosswordGrid tbody tr td{ width: 1.5em; height:
1.5em; text-align: center; vertical-align: middle; font-size: 140%; padding: 1px; margin: 0px; border-style: solid;
border-width: 1px; border-color: #000000; color: #000000; } table.CrosswordGrid span{ color: #000000; }
table.CrosswordGrid td.BlankCell{ background-color: #000000; color: #000000; } table.CrosswordGrid
td.LetterOnlyCell{ text-align: center; vertical-align: middle; background-color: #ffffff; color: #000000; font-weight:
bold; } table.CrosswordGrid td.NumLetterCell{ text-align: left; vertical-align: top; background-color: #ffffff; color:
#000000; padding: 1px; font-weight: bold; } .NumLetterCellText{ cursor: pointer; color: #000000; } .GridNum{
vertical-align: super; font-size: 66%; font-weight: bold; text-decoration: none; color: #000000; } .GridNum:hover,
.GridNum:visited{ color: #000000; } table#Clues{ margin: auto; vertical-align: top; } table#Clues td{ vertical-align:
top; } table.ClueList{ margin: auto; } td.ClueNum{ text-align: right; font-weight: bold; vertical-align: top; } td.Clue{
text-align: left; } div#ClueEntry{ text-align: left; margin-bottom: 1em; } /* Keypad styles */ div.Keypad{ text-align:
center; display: none; /* initially hidden, shown if needed */ margin-bottom: 0.5em; } div.Keypad button{ font-family:
Geneva,Arial; font-size: 120%; background-color: #ffffff; color: #000000; width: 2em; } /* JQuiz styles */
div.QuestionNavigation{ text-align: center; } .QNum{ margin: 0em 1em 0.5em 1em; font-weight: bold;
vertical-align: middle; } textarea{ font-family: Geneva,Arial; } .QuestionText{ text-align: left; margin: 0px; font-size:
```

```

100%; } .Answer{ font-size: 120%; letter-spacing: 0.1em; } .PartialAnswer{ font-size: 120%; letter-spacing: 0.1em;
color: #000066; } .Highlight{ color: #000000; background-color: #ffff00; font-weight: bold; font-size: 120%; }
ol.QuizQuestions{ text-align: left; list-style-type: none; } li.QuizQuestion{ padding: 1em; border-style: solid;
border-width: 0px 0px 1px 0px; } ol.MCAnswers{ text-align: left; list-style-type: upper-alpha; padding: 1em; }
ol.MCAnswers li{ margin-bottom: 1em; } ol.MSelAnswers{ text-align: left; list-style-type: lower-alpha; padding:
1em; } div.ShortAnswer{ padding: 1em; } .FuncButton { text-align: center; border-style: solid; border-left-color:
#eeeeff; border-top-color: #eeeeff; border-right-color: #6e6e7f; border-bottom-color: #6e6e7f; color: #000066;
background-color: #DDDDFF; border-width: 2px; padding: 3px 6px 3px 6px; cursor: pointer; } .FuncButtonUp {
color: #DDDDFF; text-align: center; border-style: solid; border-left-color: #eeeeff; border-top-color: #eeeeff;
border-right-color: #6e6e7f; border-bottom-color: #6e6e7f; background-color: #000066; color: #DDDDFF;
border-width: 2px; padding: 3px 6px 3px 6px; cursor: pointer; } .FuncButtonDown { color: #DDDDFF; text-align:
center; border-style: solid; border-left-color: #6e6e7f; border-top-color: #6e6e7f; border-right-color: #eeeeff;
border-bottom-color: #eeeeff; background-color: #000066; color: #DDDDFF; border-width: 2px; padding: 3px 6px
3px 6px; cursor: pointer; } /*BeginNavBarStyle*/ div.NavButtonBar{ background-color: #004080; text-align: center;
margin: 2px 0px 2px 0px; clear: both; font-size: 100%; } .NavButton { border-style: solid; border-left-color:
#7f9fbf; border-top-color: #7f9fbf; border-right-color: #002040; border-bottom-color: #002040; background-color:
#004080; color: #ffffff; border-width: 2px; cursor: pointer; } .NavButtonUp { border-style: solid; border-left-color:
#7f9fbf; border-top-color: #7f9fbf; border-right-color: #002040; border-bottom-color: #002040; color: #004080;
background-color: #ffffff; border-width: 2px; cursor: pointer; } .NavButtonDown { border-style: solid;
border-left-color: #002040; border-top-color: #002040; border-right-color: #7f9fbf; border-bottom-color: #7f9fbf;
color: #004080; background-color: #ffffff; border-width: 2px; cursor: pointer; } /*EndNavBarStyle*/ div.CardStyle
{ position: absolute; font-family: Geneva,Arial; font-size: 100%; padding: 5px; border-style: solid; border-width:
1px; color: #000066; background-color: #DDDDFF; left: -50px; top: -50px; overflow: visible; } .rtl{ text-align: right;
font-size: 140%; } <![CDATA[ <!-- function Client(){ //if not a DOM browser, hopeless this.min = false; if
(document.getElementById){this.min = true;}; this.ua = navigator.userAgent; this.name = navigator.appName;
this.ver = navigator.appVersion; //Get data about the browser this.mac = (this.ver.indexOf('Mac') != -1); this.win =
(this.ver.indexOf('Windows') != -1); //Look for Gecko this.gecko = (this.ua.indexOf('Gecko') > 1); if (this.gecko){
this.geckoVer = parseInt(this.ua.substring(this.ua.indexOf('Gecko')+6, this.ua.length)); if (this.geckoVer <
20020000){this.min = false; } //Look for Firebird this.firebird = (this.ua.indexOf('Firebird') > 1); //Look for Safari
this.safari = (this.ua.indexOf('Safari') > 1); if (this.safari){ this.gecko = false; } //Look for IE this.ie =
(this.ua.indexOf('MSIE') > 0); if (this.ie){ this.ieVer = parseFloat(this.ua.substring(this.ua.indexOf('MSIE')+5,
this.ua.length)); if (this.ieVer < 5.5){this.min = false; } //Look for Opera this.opera = (this.ua.indexOf('Opera') > 0);
if (this.opera){ this.operaVer = parseFloat(this.ua.substring(this.ua.indexOf('Opera')+6, this.ua.length)); if
(this.operaVer < 7.04){this.min = false; } if (this.min == false){ alert('Your browser may not be able to handle this
page. '); } //Special case for the horrible ie5mac this.ie5mac = (this.ie&&this.mac&&(this.ieVer<6)); } var C = new
Client(); //for (prop in C){ // alert(prop + ': ' + C[prop]); // } //CODE FOR HANDLING NAV BUTTONS AND
FUNCTION BUTTONS //[/strNavBarJS] function NavBtnOver(Btn){ if (Btn.className !=
'NavButtonDown'){Btn.className = 'NavButtonUp';} } function NavBtnOut(Btn){ Btn.className = 'NavButton'; }
function NavBtnDown(Btn){ Btn.className = 'NavButtonDown'; } //[/strNavBarJS] function FuncBtnOver(Btn){ if
(Btn.className != 'FuncButtonDown'){Btn.className = 'FuncButtonUp';} } function FuncBtnOut(Btn){
Btn.className = 'FuncButton'; } function FuncBtnDown(Btn){ Btn.className = 'FuncButtonDown'; } function
FocusAButton(){ if (document.getElementById('CheckButton1') != null){
document.getElementById('CheckButton1').focus(); } else{ if (document.getElementById('CheckButton2') != null){
document.getElementById('CheckButton2').focus(); } else{
document.getElementsByTagName('button')[0].focus(); } } } //CODE FOR HANDLING DISPLAY OF POPUP
FEEDBACK BOX var topZ = 1000; function ShowMessage(Feedback){ var Output = Feedback + '
'; document.getElementById('FeedbackContent').innerHTML = Output; var FDiv =
document.getElementById('FeedbackDiv'); topZ++; FDiv.style.zIndex = topZ; FDiv.style.top =
TopSettingWithScrollOffset(30) + 'px'; FDiv.style.display = 'block'; ShowElements(false, 'input');
ShowElements(false, 'select'); ShowElements(false, 'object'); //Focus the OK button
setTimeout("document.getElementById('FeedbackOKButton').focus()", 50); // } //Added code to work around

```

```

Quicktime bug for 6.0.4.24. function ShowElements(Show, TagName){ //Special for IE bug -- hide all the form
elements that will show through the popup if ((C.ie)||((TagName == 'object'))){ var Els =
document.getElementsByTagName(TagName); for (var i=0; i<Els.length; i++){ if (Show == true){
Els[i].style.visibility = 'visible'; } else{ Els[i].style.visibility = 'hidden'; } } } function HideFeedback(){
document.getElementById('FeedbackDiv').style.display = 'none'; ShowElements(true, 'input'); ShowElements(true,
'select'); ShowElements(true, 'object'); if (Finished == true){ Finish(); } } //GENERAL UTILITY FUNCTIONS AND
VARIABLES //PAGE DIMENSION FUNCTIONS function PageDim(){ //Get the page width and height this.W = 600;
this.H = 400; this.W = document.getElementsByTagName('body')[0].clientWidth; this.H =
document.getElementsByTagName('body')[0].clientHeight; } var pg = null; function GetPageXY(EI) { var XY = {x: 0,
y: 0}; while(EI){ XY.x += EI.offsetLeft; XY.y += EI.offsetTop; EI = EI.offsetParent; } return XY; } function
GetScrollTop(){ if (document.documentElement && document.documentElement.scrollTop){ return
document.documentElement.scrollTop; } else{ if (document.body){ return document.body.scrollTop; } else{
return window.pageYOffset; } } } function GetViewportHeight(){ if (window.innerHeight){ return
window.innerHeight; } else{ return document.getElementsByTagName('body')[0].clientHeight; } } function
TopSettingWithScrollOffset(TopPercent){ var T = Math.floor(GetViewportHeight() * (TopPercent/100)); return
GetScrollTop() + T; } //CODE FOR AVOIDING LOSS OF DATA WHEN BACKSPACE KEY INVOKES history.back()
var InTextBox = false; function SuppressBackspace(e){ if (InTextBox == true){return;} if (C.ie) { thisKey =
window.event.keyCode; } else { thisKey = e.keyCode; } var Suppress = false; if (thisKey == 8) { Suppress =
true; } if (Suppress == true){ if (C.ie){ window.event.returnValue = false; window.event.cancelBubble = true; }
else{ e.preventDefault(); } } } if (C.ie){ document.attachEvent('onkeydown',SuppressBackspace);
window.attachEvent('onkeydown',SuppressBackspace); } else{ if (window.addEventListener){
window.addEventListener('keypress',SuppressBackspace,false); } } function ReduceItems(InArray,
ReduceToSize){ var ItemToDump=0; var j=0; while (InArray.length > ReduceToSize){ ItemToDump =
Math.floor(InArray.length*Math.random()); InArray.splice(ItemToDump, 1); } } function Shuffle(InArray){ var Num;
var Temp = new Array(); var Len = InArray.length; var j = Len; for (var i=0; i<Len; i++){ Temp[i] = InArray[i]; }
for (i=0; i<Len; i++){ Num = Math.floor(j * Math.random()); InArray[i] = Temp[Num]; for (var k=Num; k < (j-1);
k++){ Temp[k] = Temp[k+1]; } j--; } return InArray; } function WriteToInstructions(Feedback) {
document.getElementById('InstructionsDiv').innerHTML = Feedback; } function EscapeDoubleQuotes(InString){
return InString.replace(/"/g, "") } function TrimString(InString){ var x = 0; if (InString.length != 0) {
while ((InString.charAt(InString.length - 1) == '\u0020' || (InString.charAt(InString.length - 1) == '\u000A' ||
(InString.charAt(InString.length - 1) == '\u000D'))){ InString = InString.substring(0, InString.length - 1)
} while ((InString.charAt(0) == '\u0020' || (InString.charAt(0) == '\u000A' || (InString.charAt(0) ==
'\u000D'))){ InString = InString.substring(1, InString.length) } while
(InString.indexOf(' ') != -1) { x = InString.indexOf(' ') InString = InString.substring(0, x)
+ InString.substring(x+1, InString.length) } return InString; } else { return "";
} } } function FindLongest(InArray){ if (InArray.length < 1){return -1;} var Longest = 0; for (var i=1;
i<InArray.length; i++){ if (InArray[i].length > InArray[Longest].length){ Longest = i; } } return Longest; }
//UNICODE CHARACTER FUNCTIONS function IsCombiningDiacritic(CharNum){ var Result = (((CharNum >=
0x0300)&&(CharNum <= 0x370))||((CharNum >= 0x20d0)&&(CharNum <= 0x20ff))); Result = Result || (((CharNum
>= 0x3099)&&(CharNum <= 0x309a))||((CharNum >= 0xfe20)&&(CharNum <= 0xfe23))); return Result; } function
IsCJK(CharNum){ return ((CharNum >= 0x3000)&&(CharNum < 0xd800)); } //SETUP FUNCTIONS //BROWSER
WILL REFILL TEXT BOXES FROM CACHE IF NOT PREVENTED function ClearTextBoxes(){ var NList =
document.getElementsByTagName('input'); for (var i=0; i<NList.length; i++){ if ((NList[i].id.indexOf('Guess') >
-1)||((NList[i].id.indexOf('Gap') > -1)){ NList[i].value = ""; } if (NList[i].id.indexOf('Chk') > -1){ NList[i].checked = "";
} } } //EXTENSION TO ARRAY OBJECT function Array_IndexOf(Input){ var Result = -1; for (var i=0; i<this.length;
i++){ if (this[i] == Input){ Result = i; } } return Result; } Array.prototype.indexOf = Array_IndexOf; //IE HAS
RENDERING BUG WITH BOTTOM NAVBAR function RemoveBottomNavBarForIE(){ if
((C.ie)&&(document.getElementById('Reading') != null)){ if (document.getElementById('BottomNavBar') != null){
document.getElementById('TheBody').removeChild(document.getElementById('BottomNavBar')); } } }
//HOTPOTNET-RELATED CODE var HPNStartTime = (new Date()).getTime(); var SubmissionTimeout = 30000; var
Detail = ""; //Global that is used to submit tracking data function Finish(){ //If there's a form, fill it out and submit it if
(document.store != null){ Frm = document.store; Frm.starttime.value = HPNStartTime; Frm.endtime.value = (new

```

```

Date()).getTime(); Frm.mark.value = Score; Frm.detail.value = Detail; Frm.submit(); } } //JCLOZE CORE
JAVASCRIPT CODE function ItemState(){ this.ClueGiven = false; this.HintsAndChecks = 0;
this.MatchedAnswerLength = 0; this.ItemScore = 0; this.AnsweredCorrectly = false; this.Guesses = new Array();
return this; } var Feedback = ""; var Correct = 'Bravo! Bien jou\u00e9.'; var Incorrect = 'Une partie de la r\u00e9ponse
est correcte. Les erreurs ont \u00e9t\u00e9 effac\u00e9es.'; var GiveHint = 'La lettre suivante a \u00e9t\u00e9
ajout\u00e9e.'; var CaseSensitive = true; var YourScoreIs = 'Votre score est de.'; var Finished = false; var Locked =
false; var Score = 0; var CurrentWord = 0; var Guesses = ""; var TimeOver = false; I = new Array(); I=new Array(); I[0]
= new Array(); I[0][1] = new Array(); I[0][1][0] = new Array(); I[0][1][0][0] =
'\u006e\u0065\u0075\u0072\u0061\u006d\u0069\u006e\u0069\u0064\u0061\u0073\u0065'; I[0][1][1] = new Array();
I[0][1][1][0] = '\u006e\u0065\u0075\u0072\u0061\u006d\u0069\u006e\u0069\u0064\u0061\u0073\u0065'; I[0][2] = 'n';
I[1] = new Array(); I[1][1] = new Array(); I[1][1][0] = new Array(); I[1][1][0][0] =
'\u0068\u00e9\u006d\u0067\u0067\u006c\u0075\u0074\u0069\u006e\u0069\u006e\u0065'; I[1][1][1] = new
Array(); I[1][1][1][0] =
'\u0068\u00e9\u006d\u0067\u0067\u006c\u0075\u0074\u0069\u006e\u0069\u006e\u0065'; I[1][1][2] = new
Array(); I[1][1][2][0] =
'\u0068\u0065\u006d\u0061\u0067\u0067\u006c\u0075\u0074\u0069\u006e\u0069\u006e\u0065'; I[1][2] = 'h'; State
= new Array(); function StartUp(){ RemoveBottomNavBarForIE(); //Show a keypad if there is one (added bugfix for
6.0.4.12) if (document.getElementById('CharacterKeypad') != null){
document.getElementById('CharacterKeypad').style.display = 'block'; } var i = 0; State.length = 0; for (i=0;
i<I.length; i++){ State[i] = new ItemState(); } ClearTextBoxes(); StartTimer(); } function ShowClue(ItemNum){
if (Locked == true){return;} State[ItemNum].ClueGiven = true; ShowMessage(I[ItemNum][2]); } function
SaveCurrentAnswers(){ var Ans = ""; for (var i=0; i<I.length; i++){ Ans = GetGapValue(i); if ((Ans.length >
0)&&(Ans != State[i].Guesses[State[i].Guesses.length-1])){ State[i].Guesses[State[i].Guesses.length] = Ans; } } }
function CompileGuesses(){ var F = document.getElementById('store'); if (F != null){ var Temp = '<?xml
version="1.0"?>'; var GapLabel = ""; for (var i=0; i + GapLabel + "; Temp += 'student-responses' + GapLabel + ";
Temp += 'JClozeStudentResponses'; for (var j=0; j<State[i].Guesses.length; j++){ if (j>0){Temp += '|'; Temp +=
State[i].Guesses[j] + '|'; } Temp += "; } Temp += "; Detail = Temp; } } function CheckAnswers(){ if (Locked ==
true){return;} SaveCurrentAnswers(); var AllCorrect = true; //Check each answer for (var i = 0; i<I.length; i++){ if
(State[i].AnsweredCorrectly == false){ //If it's right, calculate its score if (CheckAnswer(i, true) > -1){ var TotalChars =
GetGapValue(i).length; State[i].ItemScore = (TotalChars-State[i].HintsAndChecks)/TotalChars; if (State[i].ClueGiven
== true){State[i].ItemScore /= 2; } if (State[i].ItemScore < 0 ){State[i].ItemScore = 0; } State[i].AnsweredCorrectly = true;
//Drop the correct answer into the page, replacing the text box SetCorrectAnswer(i, GetGapValue(i)); } else{
//Otherwise, increment the hints for this item, as a penalty State[i].HintsAndChecks++; //then set the flag AllCorrect =
false; } } //Calculate the total score var TotalScore = 0; for (i=0; i
'; } Output += YourScoreIs + ' ' + TotalScore + '%.
'; if (AllCorrect == false){ Output += '
' + Incorrect; } ShowMessage(Output); setTimeout('WriteToInstructions(Output)', 50); Score = TotalScore;
CompileGuesses(); if ((AllCorrect == true)||((Finished == true))){ window.clearInterval(Interval); TimeOver = true;
Locked = true; Finished = true; setTimeout('Finish()', SubmissionTimeout); } } function TrackFocus(BoxNumber){
CurrentWord = BoxNumber; InTextBox = true; } function LeaveGap(){ InTextBox = false; } function
CheckBeginning(Guess, Answer){ var OutString = ""; var i = 0; var UpperGuess = ""; var UpperAnswer = ""; if
(CaseSensitive == false) { UpperGuess = Guess.toUpperCase(); UpperAnswer = Answer.toUpperCase(); } else {
UpperGuess = Guess; UpperAnswer = Answer; } while (UpperGuess.charAt(i) == UpperAnswer.charAt(i)) { OutString
+= Guess.charAt(i); i++; } OutString += Answer.charAt(i); return OutString; } function GetGapValue(GNum){ var
RetVal = ""; if ((GNum<0)||((GNum>=I.length))){return RetVal; } if (document.getElementById('Gap' + GNum) != null){
RetVal = document.getElementById('Gap' + GNum).value; RetVal = TrimString(RetVal); } else{ RetVal =
State[GNum].Guesses[State[GNum].Guesses.length-1]; } return RetVal; } function SetGapValue(GNum, Val){ if
((GNum<0)||((GNum>=I.length))){return;} if (document.getElementById('Gap' + GNum) != null){
document.getElementById('Gap' + GNum).value = Val; document.getElementById('Gap' + GNum).focus(); } } function
SetCorrectAnswer(GNum, Val){ if ((GNum<0)||((GNum>=I.length))){return;} if (document.getElementById('GapSpan' +
GNum) != null){ document.getElementById('GapSpan' + GNum).innerHTML = Val; } } function FindCurrent() { var x =
0; FoundCurrent = -1; //Test the current word: //If its state is not set to already correct, check the word. if

```

```

(State[CurrentWord].AnsweredCorrectly == false){ if (CheckAnswer(CurrentWord, false) < 0){ return CurrentWord; } }
x=CurrentWord + 1; while (x<l.length){ if (State[x].AnsweredCorrectly == false){ if (CheckAnswer(x, false) < 0){ return
x; } } x++; } x = 0; while (x<CurrentWord){ if (State[x].AnsweredCorrectly == false){ if (CheckAnswer(x, false) < 0){
return x; } } x++; } return FoundCurrent; } function CheckAnswer(GapNum, MarkAnswer){ var Guess =
GetGapValue(GapNum); var UpperGuess = ""; var UpperAnswer = ""; if (CaseSensitive == false){ UpperGuess =
Guess.toUpperCase(); } else{ UpperGuess = Guess; } var Match = -1; for (var i = 0; i<l[GapNum][1].length; i++){ if
(CaseSensitive == false){ UpperAnswer = l[GapNum][1][i][0].toUpperCase(); } else{ UpperAnswer =
l[GapNum][1][i][0]; } if (TrimString(UpperGuess) == UpperAnswer){ Match = i; if (MarkAnswer == true){
State[GapNum].AnsweredCorrectly = true; } } } return Match; } function GetHint(GapNum){ Guess =
GetGapValue(GapNum); if (CheckAnswer(GapNum, false) > -1){return ""} RightBits = new Array(); for (var i=0; i
<l[GapNum][1].length; i++){ RightBits[i] = CheckBeginning(Guess, l[GapNum][1][i][0]); } var RightOne =
FindLongest(RightBits); var Result = l[GapNum][1][RightOne][0].substring(0,RightBits[RightOne].length); //Add
another char if the last one is a space if (Result.charAt(Result.length-1) == ' '){ Result =
l[GapNum][1][RightOne][0].substring(0,RightBits[RightOne].length+1); } return Result; } function ShowHint(){ if
(document.getElementById('FeedbackDiv').style.display == 'block'){return;} if (Locked == true){return;} var CurrGap =
FindCurrent(); if (CurrGap < 0){return;} var HintString = GetHint(CurrGap); if (HintString.length > 0){
SetGapValue(CurrGap, HintString); State[CurrGap].HintsAndChecks += 1; } ShowMessage(GiveHint); } function
TypeChars(Chars){ var CurrGap = FindCurrent(); if (CurrGap < 0){return;} if (document.getElementById('Gap' +
CurrGap) != null){ SetGapValue(CurrGap, document.getElementById('Gap' + CurrGap).value + Chars); } } function
TimesUp() { document.getElementById('Timer').innerHTML = 'Le temps imparti est \u00e9coulu\u00e9 !'; TimeOver =
true; Finished = true; CheckAnswers(); Locked = true; } //CODE FOR HANDLING TIMER //Timer code var Seconds =
45; var Interval = null; function StartTimer(){ Interval = window.setInterval('DownTime()',1000);
document.getElementById('TimerText').style.display = 'inline'; } function DownTime(){ var ss = Seconds % 60; if
(ss<10){ ss='0' + ss + "; } var mm = Math.floor(Seconds / 60); if (document.getElementById('Timer') == null){ return; }
document.getElementById('TimerText').innerHTML = mm + ':' + ss; if (Seconds < 1){ window.clearInterval(Interval);
TimeOver = true; TimesUp(); } Seconds--; } //--» //]]>

```

# Signification de l'expression Grippe A H1 N1

## Texte à trous.

Complétez le texte. Cliquez ensuite sur "correction". Vous pouvez utiliser "Aide" pour obtenir une lettre indice. Vous pouvez aussi cliquer sur "Indice" pour obtenir un indice. Vous perdrez des points si vous utilisez "Aide" ou "Indice"!  
 <!-- Correction Aide -->

Question : Trouvez pour Lundi 07 09 2009 la signification des initiales H et N dans l'expression grippe A H1N1.

Réponse :

Le H et le N correspondent aux initiales de deux protéines présentes dans la capsid du Virus. Ces protéines sont la Hemagglutinine et la Neuraminidase. Les indices correspondent aux variants antigéniques de ces protéines.

L'image du portfolio illustre au Microscope Electronique à Transmission l'aspect de la particule virale. (taille 80 - 120 nm )

Vous pouvez consulter la page placée en lien en bas de cet article pour découvrir les caractéristiques détaillées du virus.

à ç é ê è ë ù ô À Ç É Ê Correction Aide

OK