

Blue are the Arizona Skies

A Memoir in English and Spanish

Azul es el cielo de Arizona

Memorias en inglés y español

Tom Cole

Índice/Table of Contents

Agradecimientos	i
Acknowledgments	i
Introducción	ii
Introduction.....	iii
1. Of Such Is the Essence of Nightmares.....	1
2. The Bum on the Tracks	3
3. Fun with Code! How to Program a Computer	4
Del mismo autor/By the Same Author	17

Agradecimientos

Acknowledgments

Introducción

Introduction

1. Of Such Is the Essence of Nightmares

I still cannot believe that we ever did such a thing. But we did. Just the same, the memory of it is hazy as though perhaps I have driven it from my mind in an effort at self preservation, turned my mind away from the very thought of it—the way you avert your eyes at the sight of a bad accident or unconsciously break your unsettling reverie of it.

I ask myself how such an action could have actually been put into effect and why, I also ask myself, is it only now, years later, that I have suddenly come to wonder how it could possibly have been that among us there was not a single clear head, no voice of reason to halt our madness in its tracks?

No, dear reader we didn't kill anyone and no animal was harmed in the execution of our folly. In addition, I must add, indeed, that by chance our lunacy was of no consequence in the end; a kind fate delivered us from the cataclysm that we willfully tempted, from the dizzying fall from the crag upon whose razor's edge we dared to dance our mad jig.

And so I now must tell you what we did. It is this: we left two oscillating space heaters on inside the house.

Yes, I am aware that I disappoint you, for this may not seem like a matter of great importance. But consider this: the house was abandoned in the northern Arizona woods with the heaters set to high and unattended for months—*months—space* heaters of the kind that have set endless fires and burnt to the ground countless structures and taken lives without number. *Unattended*. Infernal machines raging and blazing full tilt in a house in the forest a three-hour drive from the closest of us.

Strangely, as I reflect upon this, I cannot remember why it was even done or imagine a single sane reason why it should have been. Oh, we didn't want the pipes to freeze during the Flagstaff winter. I know that, but why was it that we didn't do this time what we had always done before: winterize the house by draining the pipes and pouring a little antifreeze in the traps and toilets?

I don't know, and I am commencing now to think it's best to cease this reflection altogether. To thus conclude, I will only ask how any of us could have slept a single night with the knowledge we shared for of such is the essence of nightmares.

2. The Bum on the Tracks

3. Fun with Code! Programing a Computer

In the early nineties our department at Arizona State University wanted to create its own webpage. We enlisted the university IT to help us, but things just weren't getting done. The art and wizardry of coding HTML seemed to be steeped in mysticism and only the IT people were privy to its abstruse riddles. Only they could unravel its deep mysteries, decipher its secrets, plumb its inscrutable depths.

Since IT was being so pokey, I got a single sheet of paper with HTML examples and was able to make our page myself in no time at all.

They were furious!

The point is that while computer programming languages can get very complicated, for the most part, they are easy enough for a 10-year-old to use. They are also a lot of fun.

How? Well, aside from practical projects of mine like my grammar programs and my "super smokin' bird processor," I have also written a Computer Psychiatrist Program, a program called Haiku Guru that turns your nouns and adjectives and phrases into a completely different Haiku with every click of a button, and a Hemingway Story Generator.

The computer psychiatrist program worked by employing tricks to give the appearance of artificial intelligence.

The user would type, "I'm feeling depressed," and the computer would randomly answer in any number of ways:

"You say you're feeling depressed. Why is that?" or "Depression is a common problem. Tell me more." or "Are you indeed? What do you think can be done about it?"

It isn't hard to get the program to use the user's words to imitate a real conversation.

There are lots of tricks. You can write code, for example, that waits a while and if the user hasn't typed "my mom" or "mother," it will respond with, "You haven't said a single word about your mother. Is there a reason for that?"

Well, you can see that you are only limited by your imagination.

I'd now like to explain how the Hemingway Story Generator functions and show you the actual code that makes it work. Afterwards, you'll know some computer programming that you likely didn't know before. I'll keep it short and simple while endeavoring to keep your interest.

Let's start with something called a *variable*. A variable is simply a word or a single letter

that represents data. Just think of a variable as a sheet of notebook paper with words and lines on it. If you create a variable named, say, MyVariable and you tell the computer to put "It was 1942." into MyVariable, (or into line 1 of MyVariable) it will become:

It was December 14, 1942.

If you then tell the computer to put "Ten dead soldiers lay in the road." into line 2 of MyVariable, it will become:

It was 1942.

Ten dead soldiers lay in the road.

Darn, that looks like a list. Let's try again. You want it to look like a paragraph in a novel so don't put anything into line 2; instead use the word AFTER. (Remember MyVariable is "It was 1942." Tell the computer this:

Put " Ten dead soldiers lay in the road." *after* MyVariable." (or after line 1)

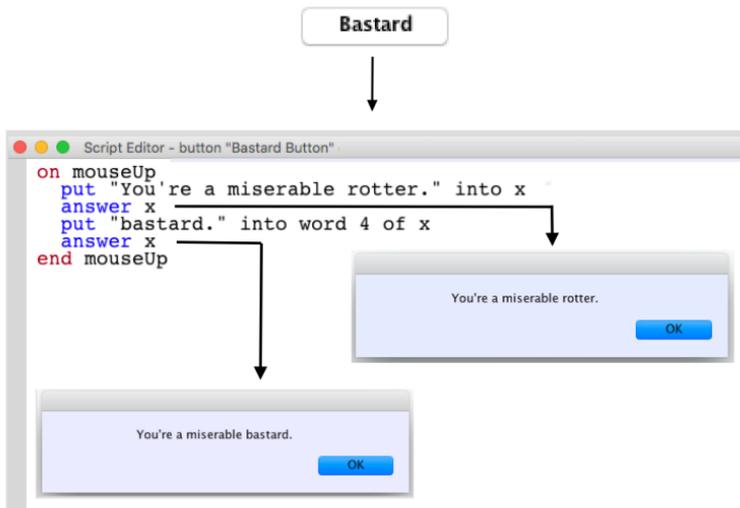
Now line 1 of MyVariable will be:

It was 1942. Ten dead soldiers lay in the road.

It's really child's play.

Here's how the screen might look when you're programming. The following code uses a single letter as a variable. I chose "x" to use. Since Hemingway's characters always curse, I've included some typical Hemingway bad language.

Read this. It's the code that's inside a button you click. I'll call it "the Bastard Button." Watch what it does. Again read the code:



See? You've learned that "on mouseup" means "when you click the button." You see once again the word "put" that we used before and you also now know that "answer" will bring up one of those message boxes.

The more astute of my readers will notice that "put into" *replaces* the word "rotter" with "bastard." The very most astute will already

know that using "put after" will *insert* a word, so

```
put " bloody " after word 2 of x
```

will give you:

You're a bloody miserable bastard.

Let's jazz it up just once more with

```
Put "!" into the last character  
of x
```

and you'll, of course, replace the period and get: "You're a bloody miserable bastard!"

You have now already learned enough about manipulating words and data with code that most of the story generator can be made. There are two more items to learn for this, but we'll wait.

Let's start. For instructional purposes, we'll make our first draft of the story just two sentences. Here they are.

```
It was a rainy day in 1932 and Johnny said,  
"You're a bloody miserable bastard!"
```

Since you're the programmer, you can choose to change the insult at the end to any number of things. But let's just use the two we had before. We'll use a pull-down menu and create a field named "Insults" to put them in.

```
You're a bloody miserable bastard!  
You're a bloody miserable rotter!
```

Our story will be put in a variable called `thestory` which will be continuously updated.

We'll use the pull-down menu and create a button we'll name "Create Story." To include the insult we just write some code in the button. You put the insult in a variable called `insult`.

(Variables embedded in text need an ampersand as you see.)

Create Story



```
Put "It was a rainy day in 1932  
and Johnny said, "&insult&" into  
thestory
```

So `thestory` is now

```
It was a rainy day in 1932 and Johnny said,  
"You're a bloody miserable rotter!"
```

As you see, I chose "rotter" instead of "bastard" but we want the *program* not me to

choose an insult and at *random*. Here's what you can do.

```
put random (2) into x
put line x of field "insults" into
insult
```

Now half the time it will be rotter and half the time bastard. Of course, you'll want to put zillions more insults in the field to add to the variety. And you won't limit yourself to insults. You will want to change "a mild day" to "a rainy October evening" or "one hell of a sweltering summer afternoon." You would make a whole list of alternates like that and put them in a field and create a variable with a name like `timeofday`. Then, you'd write the code:

Create Story



```
Put "It was "&timeofday&" in 1932
and Johnny said, "&insult&" into
thestory
```

The story will now be something like:

It was one hell of a sweltering summer afternoon in 1932 and Johnny said, "You're a bloody miserable bastard!"

Now, let's allow the user to participate. We will let him or her choose the year and the person's first name. We'll make fields for them to type their choices in.



Jack 1969

With new variables we'll call `firstname` and `theyear`, the code for the story will be:

Create Story



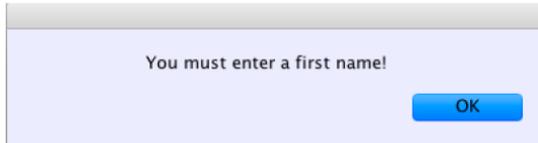
```
Put "It was "&timeofday&" in
"&theyear&" and "&firstname&" said,
"&insult&" into thestory
```

It was a rainy October evening in 1969 and
Jack said, "You're a bloody miserable rotter!"

Already the program is generating a number of different, random versions. All you need to do is do the same thing for the rest of the story and add more insults, time of days, and so on. The more you add, the more fun and impressive the results of the generator will be.

Let's finish, this rough version of the program by making sure the user has filled in all the words the story requires. We don't want the user to click the Create Story button if they

haven't chosen a year or a first name, so we'll use an "if/then" clause in the first part of the Create Story button code. You want something like this:



And as you know the code for this is the word *answer*. Thus, you will use it in the Create story button.

Create Story



On mouseup

```
if field "first name" is empty
then answer "You must enter a first
name!"
```

```
  exit mouseup
```

```
end if
```

```
if field "The Year" is empty then
answer "You must enter a year!"
```

```
  exit mouseup
```

```
end if
```

```
Put "It was "&timeofday&" in
"&theyear&" and "&firstname&" said,
"&insult&" into thestory
```

Put the story into field "Finished Story"

end mouseup

The last line of code puts the whole story into a field for the user to read, print, or copy.

The screenshot shows a web browser window with the title "Hemingway Story Generator New (1) *". The page has a grey background and a red header "HEMINGWAY STORY GENERATOR". On the left, there are ten labels for user input: "Type the first name of character 1 in the space:", "Type the last name of character 1 in the space:", "Type the first name of character 2 in the space:", "Type the last name of character 2 in the space:", "Type the name of the city in the space:", "Type the name of the state in the space:", "Type month in the space:", "Type the year in the space:", "Type the drink the people are drinking:", and "Type the name of the drinking establishment:". To the right of these labels is a vertical stack of ten white input boxes with black borders, containing the following text: "Jack", "Linton", "Hank", "Harris", "Hooper", "Nebraska", "May", "1948", "Burgundy", and "Jose's". Below the input boxes is a large white text area containing a generated story. At the bottom of the page are three buttons: "Create Story", "Print", and "Copy Story".

Type the first name of character 1 in the space:	Jack
Type the last name of character 1 in the space:	Linton
Type the first name of character 2 in the space:	Hank
Type the last name of character 2 in the space:	Harris
Type the name of the city in the space:	Hooper
Type the name of the state in the space:	Nebraska
Type month in the space:	May
Type the year in the space:	1948
Type the drink the people are drinking:	Burgundy
Type the name of the drinking establishment:	Jose's

It was a unremarkable May day in 1948. Hooper was a whistlestop of a town and Jack Linton and Hank Harris sat on barstools at Jose's. It was their favorite bistro/bar.

'Harris, I'm not even going to guess what you're drinking.' Jack muttered.

'The usual, Burgundy!' said Hank.

They ordered. When the drinks came, Jack said, 'You filthy good-for-nothing louse, Harris!'

'Oh, now you've gone and done it. You'll have one less Christmas gift this year.'

'Don't talk anymore. Pretend you're a human being and drink your Burgundy'.

Create Story Print Copy Story

Above you see the quick version I just did. The user has ten choices to input and the program changes seven other items, each one of

which could be any of ten in its field, so there are thousands of variations already. Of course, the ultimate goal of this kind of programming would be to have every single word in the story represented by a variable that could have a hundred incarnations. But for now, I just have a simple story generator.

Here are three stories that were generated by clicking the Create Story button.

I

It was a unremarkable May day in 1948. Hooper was a big, bustling town and Jack Linton and Hank Harris sat on barstools at Jose's. It was their favorite saloon.

'Harris, What're you having?' Jack muttered.

'The usual, Burgundy!' said Hank.

They ordered. When the drinks came, Jack said, 'When you drink that shit you look like a great big ape, Harris!'

'You're a mind reader. I was just thinking the exact same thing about you.'

'Oh, put a sock in it, fool and drink your Burgundy'.

II

It was a dreary May day in 1948. Hooper was a fool's paradise of a town and Jack Linton and Hank Harris sat on barstools at Jose's. It was their favorite salt lick.

'Harris, I'm not even going to guess what you're drinking.' Jack muttered.

'The usual, Burgundy!' said Hank.

They ordered. How's it been going, jackass? Oops! Harris I mean!'

'Shoot. And I was just going to kiss you on the lips.'

'Shut your big mouth and drink your Burgundy'.

III

It was a rainy May day in 1948. Hooper was a seedy little town and Jack Linton and Hank Harris sat on barstools at Jose's. It was their favorite gin joint.

'Name your poison.' Jack muttered.

'The usual, Burgundy!' said Hank.

They ordered. When the drinks came, Jack said, 'You're a miscreant, a full-bore son-of-a-bitch, Harris!'

'I guess this means we're not getting married anymore.'
'Can it, rum dummy and drink your Burgundy.'

Del mismo autor/By the Same Author

LIBROS/BOOKS

The Grass on the Hill Grows Green/El pasto en la colina crece verde, CreateSpace, 2017.

Parece que fuera ayer/It Seems like Only Yesterday, CreateSpace, 2017.

Logbook from the Sea of Cortez, CreateSpace, 2015.

Dung, a Light Roasting of Frank Herbert's Dune, CreateSpace, 2014.

Recuerdo de Amnesia/Memory of Amnesia, CreateSpace, 2014.

Las misteriosas noches de antaño/The Mysterious Nights of yesterday, CreateSpace, 2012.

The Intersection, Seventeen Years of Bird Processing on One Street Corner of the World, CreateSpace, 2011.

The Sands of Pima Arroyo, CreateSpace, 2011.

Grammar/Writing Connections, University of Michigan Press, 2009, Ann Arbor, MI 48104.

The Preposition Book, University of Michigan Press, 2006, Ann Arbor, MI 48104.

The Article Book Revised Edition, University of Michigan Press, 2000, Ann Arbor, MI 48104.

The Article Book, Simon & Schuster/Prentice Hall/Regents, 1996, Englewood Cliffs, NJ 07632.

SOFTWARE

Rookie League Baseball, 2006, University of Michigan Press, 2006, Ann Arbor, MI 48104.

The Preposition Book Software, Preposition Pinball, Carp Derby, Diamond Mine, 2006, University of Michigan Press, 2006, Ann Arbor, MI 48104.

Fish Trek 2.0, an Adventure in Articles for Windows and Macintosh, 2003 University of Michigan Press, Ann Arbor, MI 48104.

Fish Trek, an Adventure in Articles for Windows, University of Michigan Press, 2000, Ann Arbor, MI 48104.

Fish Trek, an Adventure in Articles for Macintosh, University of Michigan Press, 2000, Ann Arbor, MI 48104.

Play Ball! for the Structure and Written Expression Part of the TOEFL, 1994, Gessler Publishing Co., Inc. 55 West 13th Street, New York, New York 10011-7958.