Module 1 - Computational Thinking

Hello and welcome back to week one for the third video lecture.

Today I want to talk about the idea of computational thinking. This is something that's going to help you in figuring out how to integrate algorithms automation and A.I. into your own editorial workflows. My goal here today is to try to demystify this term a bit and show you why it's an important and very powerful concept.

So this is Jeannette Wing. She is a professor at Columbia University now, she was a very early and strong proponent of the idea of computational thinking and this is how she defines it: "If the thought processes involved in formulating problems in their solutions so that the solutions are represented in a form that can be effectively carried out by an information processing agent."

So let's unpack this a bit more and talk more about what this really means. So, computational thinking is not about thinking like a computer. It's not about turning people into computers. It's about thinking in ways so that you can use a computer in the best way possible to solve a particular problem.

Computational thinking is really about abstraction. It's about being able to see a specific problem and then recognize that that specific problem is a part of a more abstract and general problem because this allows us to encode that problem for computer so that it can be solved over and over again using an algorithm.

There are different facets to abstraction, modeling, decomposition and parameterization that I'll talk about in more detail in a little bit, but really what abstraction enables is this idea of scale.

This is the main benefit of digital computing. So, again it's about being able to repeatedly solve a problem over and over and over again so we can talk about generating a hundred, a thousand or hundred thousand stories instead of just one story.

And again computational thinking is not about programming per se. It's about organizing problems so that a solution could be programmed. And even though it's not explicitly about program I would however say that learning programming would most likely help you along your way in terms of developing and mastering computational thinking skills.

Now you might still be wondering you know why are we talking about computational thinking? So there are these two economists from M.I.T. and they actually wrote this book The Second Machine Age which if you get a chance you should totally check that out. They bring in some economics principles and they argue that effective production is more likely to require both human and machine inputs. What that really means is creating hybrids or blends between computer and human.

And I would argue that computational thinkers people, who have mastered computational thinking will be more effective at exploiting the capabilities of automation and algorithms and in designing those future hybrid workflows that best take advantage of both computer and human.

Now I want to drill into some different specific facets of computational thinking and a bit more detail so we'll talk about modeling decomposition and parameterization. So, in terms of modeling, modeling is really important because it allows us to represent information in some simplified fashion for a computer system. You can think of modeling as creating a more dumbed down version of reality for the computer.

There's a sort of a distinct editorial process here in which you're sort of systematically thinking through you know, what do I include in my model, in my dumbed down version of reality? What do I exclude from my model? And what a way perhaps emphasize or deemphasize in my model or my representation of reality?

Again the world is far too complex in order to represent it fully for computer and so we have to model the world, we have to model reality, simplify it so that it fits within the the capabilities of a computer system. So let me give you an example of what this looks like.

So take for instance an app like Apple news. It has a personalized section called for you. Now Apple News doesn't know everything about what I want to see in my news, but they can develop a model or simplification of what I might be interested in based on my prior interactions with the app as well as based on my demographics, my interest, my location, my geography, and maybe some other things as well.

The ideas that they have a model of me. It's not a perfect representation of me it's just an estimate of who I am and what I'm interested in, but that model is very powerful in this case it lets Apple make some recommendations for me that might make what they're showing me more relevant to me as an individual.

So that's modeling. Now I want to talk about decomposition. Decomposition is about pulling apart the steps of a process. Some large processes we can think of as you know a composite of many smaller actions or smaller steps, and that's really the idea of decomposition breaking large tasks or processes down into sub sequences of smaller steps. So if you can break up a big complex task in to many smaller simpler tasks some of those simpler tasks might actually be solvable by a computer.

And of course other tasks you'll still need a human to do and they can't be automated, but by decomposing large tasks into smaller tasks we can start thinking through whether humans should do the task or whether a computer can do the task.

And then we can also think about how to recombine all of those subtasks into the larger task at the end. So, decomposition is about breaking things down, figuring out which pieces are better for human versus computer and then building backup in order to solve the bigger problem again.

So let's look at an example of this. So, consider the workflow for writing an article of data journalism for instance. You know maybe we can decompose that workflow into some steps.

So in this slide I'm showing you five sub steps so you can imagine the first step might be collecting data. Second step might be analyzing that data to identify some interesting events or outliers or trends. Then the third step might be prioritizing those insights, which are the more newsworthy or insightful or important events that you've identified in the data? Then maybe the fourth step is coming up with a narrative. So, how do a string together those insights into something compelling? How do we then write up those insights in an article or create a data that is two percent those insights? And then finally the last step might be publishing the story.

So putting it in a CMS and distributing it. So you can see I've kind of broken up this idea of writing an article of data journalism into these five steps and you could probably even imagine breaking each of these five steps into even smaller subtasks in order to just sort of keep breaking things down into smaller pieces.

The final facet I want to talk about here today is parameterization. The sort of idea of parameters is that they allow an abstract algorithm to apply in many different specific cases.

Now to illustrate this idea of parameterization I want to come back to this analogy of an algorithm as a recipe by parameterizing a recipe we can make it useful for creating a regular cake as well as for making a vegetarian version of this other cake. So you can see in this list of ingredients on the slide that one of the ingredients in the recipe is eggs. So what are eggs really to the recipe though?

There is something to keep the other ingredients adhered together in the batter. There are a binding agent now for a vegetarian version of the cake. We could use a different binding agent such as ground flax meal. So by abstracting the binding agent as a parameter for the recipe or the or the algorithm here we can use a parameter of eggs for the regular cake and a parameter of round flax meal for the vegetarian cake. So again the power here is that parameters allow algorithms to achieve many different outcomes to suit a much wider range of contexts.

One parameter gives us the regular cake on different parameter for the same algorithm gives us vegetarian cake. So, to recap we've talked about modeling, decomposition and parameterization. These are the three essential facets of computational thinking.

I would argue that if you've mastered these concepts then you've mastered most of what it means to think computationally. Again, if you really want to practice these skills it won't hurt to pick up a programming language. Some of the more popular languages for data journalism these days are a language called R and then another language called Python.

And if you want to do more web based programming you might want to pick up Javascript for that. So this concludes our video lectures for week one. We've got some questions for you and a quiz online, and I'm looking forward to seeing you in the forums.