Unit 2: Loops and Variables

Learning Objectives

- 1: The student can use computing tools and techniques to create artifacts
- 3: The student can use computing tools and techniques for creative expression.
- 4: The student can use programming as a creative tool.
- 9: The student can use models and simulations to raise and answer questions.
- 15: The student can develop an algorithm.
- 16: The student can express an algorithm in a language.
- 28: The student can analyze how computing affects communication, interaction, and cognition.
- 29: The student can connect computing with innovations in other fields.
- 30: The student can analyze the beneficial and harmful effects of computing.
- 31: The student can connect computing within economic, social, and cultural contexts.

Readings/Lectures

- Blown to Bits: Chapter 2 (http://www.bitsbook.com/wp-content/uploads/2008/12/chapter2.pdf)
- Lecture Slides 2.01: Applications that changed the world (/bjc-course/curriculum/02-loops-and-variables/readings/01-applications-that-changed-the-world-slides.pdf)
- Lecture Slides 2.02: Algorithms (/bjc-course/curriculum/02-loops-and-variables/readings/02-algorithms-slides.pdf)
- Lecture Slides 2.03: Algorithm Development Activity (/bjc-course/curriculum/02-loops-and-variables/readings/03-algorithms-activity-slides.pdf)
- Lecture Slides 2.04: Loops and Variables (/bjc-course/curriculum/02-loops-and-variables/readings/04-loops-variables-slides.pdf)
- Lecture Video 2.05: Variables (/bjc-course/curriculum/02-loops-and-variables/readings/05-looping-byob-video.mp4)
- Lecture Video 2.06: Looping (/bjc-course/curriculum/02-loops-and-variables/readings/06-variables-byob-video.mp4)

Labs/Exercises

- Lab 2.01: Variables and Looping (/bjc-course/curriculum/02-loops-and-variables/labs/01-variables-and-looping)
- Lab 2.02: Position on the Stage (/bjc-course/curriculum/02-loops-and-variables/labs/02-position-on-the-stage)
- Lab 2.03: Repeat Until Practice Worksheet (/bjc-course/curriculum/02-loops-and-variables/labs/03-practice-repeat-until.pdf)
- Lab 2.04: Regular Figures Worksheet (/bjc-course/curriculum/02-loops-and-variables/labs/04-regular-figures-activity.pdf)

Why We Lost Our Privacy, or Gave It Away

Information technology did not cause the end of privacy, any more than automotive technology caused teen sex. Technology creates opportunities and risks, and people, as individuals and as societies, decide how to live in the changed landscape of new possibilities. To understand why we have less privacy today than in the past, we must look not just at the gadgets. To be sure, we should be wary of spies and thieves, but we should also look at those who protect us and help us—and we should also take a good look in the mirror.

We are most conscious of our personal information winding up in the hands of strangers when we think about data loss or theft. Reports like the one about the British tax office have become fairly common. The theft of information about 45 million customers of TJX stores, described in Chapter 5, "Secret Bits," was even larger than the British catastrophe. In 2003, Scott Levine, owner of a mass email business named Snipermail, stole more than a billion personal information records from Acxiom. Millions of Americans are victimized by identity theft every year, at a total cost in the tens of billions of dollars annually. Many more of us harbor daily fears that just "a little bit" of our financial information has leaked out, and could be a personal time bomb if it falls into the wrong hands.

Why can't we just keep our personal information to ourselves? Why do so many other people have it in the first place, so that there is an opportunity for it to go astray, and an incentive for creative crooks to try to steal it?

We lose control of our personal information because of things we do to ourselves, and things others do to us. Of things we do to be ahead of the curve, and things we do because everyone else is doing them. Of things we do to save money, and things we do to save time. Of things we do to be safe from our enemies, and things we do because we feel invulnerable. Our loss of privacy is a problem, but there is no one answer to it, because there is no one reason why it is happening. It is a messy problem, and we first have to think about it one piece at a time.

We give away information about ourselves—voluntarily leave visible footprints of our daily lives—because we judge, perhaps without thinking about it very much, that the benefits outweigh the costs. To be sure, the benefits are many.

Saving Time

For commuters who use toll roads or bridges, the risk-reward calculation is not even close. Time is money, and time spent waiting in a car is also anxiety and

frustration. If there is an option to get a toll booth transponder, many commuters will get one, even if the device costs a few dollars up front. Cruising past the cars waiting to pay with dollar bills is not just a relief; it actually brings the driver a certain satisfied glow.

The transponder, which the driver attaches to the windshield from inside the car, is an RFID, powered with a battery so identifying information can be sent to the sensor several feet away as the driver whizzes past. The sensor can be mounted in a constricted travel lane, where a toll booth for a human toll-taker might have been. Or it can be mounted on a boom above traffic, so the driver doesn't even need to change lanes or slow down

And what is the possible harm? Of course, the state is recording the fact that the car has passed the sensor; that is how the proper account balance can be debited to pay the toll. When the balance gets too low, the driver's credit card may get billed automatically to replenish the balance. All that only makes the system better—no fumbling for change or doing anything else to pay for your travels.

The monthly bill—for the Massachusetts Fast Lane, for example—shows where and when you got on the highway—when, accurate to the second. It also shows where you got off and how far you went. Informing you of the mileage is another useful service, because Massachusetts drivers can get a refund on certain fuel taxes, if the fuel was used on the state toll road. Of course, you do not need a PhD to figure out that the state also knows when you got off the road, to the second, and that with one subtraction and one division, its computers could figure out if you were speeding. Technically, in fact, it would be trivial for the state to print the appropriate speeding fine at the bottom of the statement, and to bill your credit card for that amount at the same time as it was charging for tolls. That would be taking convenience a bit too far, and no state does it, yet.

What does happen right now, however, is that toll transponder records are introduced into divorce and child custody cases. You've never been within five miles of that lady's house? Really? Why have you gotten off the highway at the exit near it so many times? You say you can be the better custodial parent for your children, but the facts suggest otherwise. As one lawyer put it, "When a guy says, 'Oh, I'm home every day at five and I have dinner with my kids every single night,' you subpoena his E-ZPass and you find out he's crossing that bridge every night at 8:30. Oops!" These records can be subpoenaed, and have been, hundreds of times, in family law cases. They have also been used in employment cases, to prove that the car of a worker who said he was working was actually far from the workplace.

But most of us aren't planning to cheat on our spouses or our bosses, so the loss of privacy seems like no loss at all, at least compared to the time

saved. Of course, if we actually *were* cheating, we *would* be in a big hurry, and might take some risks to save a few minutes!

Saving Money

Sometimes it's money, not time, which motivates us to leave footprints. Such is the case with supermarket loyalty cards. If you do not want Safeway to keep track of the fact that you bought the 12-pack of Yodels despite your recent cholesterol results, you can make sure it doesn't know. You simply pay the "privacy tax"—the surcharge for customers not presenting a loyalty card. The purpose of loyalty cards is to enable merchants to track individual item purchases. (Item-level transactions are typically not tracked by credit card companies, which do not care if you bought Yodels instead of granola, so long as you pay the bill.) With loyalty cards, stores can capture details of cash transactions as well. They can process all the transaction data, and draw inferences about shoppers' habits. Then, if a lot of people who buy Yodels also buy Bison Brew Beer, the store's automated cash register can automatically spit out a discount coupon for Bison Brew as your Yodels are being bagged. A "discount" for you, and more sales for Safeway. Everybody wins. Don't they?

As grocery stores expand their web-based business, it is even easier for them to collect personal information about you. Reading the fine print when you sign up is a nuisance, but it is worth doing, so you understand what you are giving and what you are getting in return. Here are a few sentences of Safeway's privacy policy for customers who use its web site:

> Safeway may use personal information to provide you with newsletters, articles, product or service alerts, new product or service announcements, saving awards, event invitations, personally tailored coupons, program and promotional information and offers, and other information, which may be provided to Safeway by other companies. ... We may provide personal information to our partners and suppliers for customer support services and processing of personal information on behalf of Safeway. We may also share personal information with our affiliate companies, or in the course of an actual or potential sale, re-organization, consolidation, merger, or amalgamation of our business or businesses.

Dreary reading, but the language gives Safeway lots of leeway. Maybe you don't care about getting the junk mail. Not everyone thinks it is junk, and the

company does let you "opt out" of receiving it (although in general, few people bother to exercise opt-out rights). But Safeway has lots of "affiliates," and who knows how many companies with which it *might* be involved in a merger or sale of part of its business. Despite privacy concerns voiced by groups like C.A.S.P.I.A.N. (Consumers Against Supermarket Privacy Invasion and Numbering, `www.nocards.org`), most shoppers readily agree to have the data collected. The financial incentives are too hard to resist, and most consumers just don't worry about marketers knowing their purchases. But whenever purchases can be linked to your name, there is a record, somewhere in a huge database, of whether you use regular or super tampons, lubricated or unlubricated condoms, and whether you like regular beer or lite. You have authorized the company to share it, and even if you hadn't, the company could lose it accidentally, have it stolen, or have it subpoenaed.

Convenience of the Customer

The most obvious reason not to worry about giving information to a company is that you do business with them, and it is in your interest to see that they do their business with you better. You have no interest in whether they make more money from you, but you do have a strong interest in making it easier and faster for you to shop with them, and in cutting down the amount of stuff they may try to sell you that you would have no interest in buying. So your interests and theirs are, to a degree, aligned, not in opposition. Safeway's privacy policy states this explicitly: "Safeway Club Card information and other information may be used to help make Safeway's products, services, and programs more useful to its customers." Fair enough.

No company has been more progressive in trying to sell customers what they might want than the online store Amazon. Amazon suggests products to repeat customers, based on what they have bought before—or what they have simply looked at during previous visits to Amazon's web site. The algorithms are not perfect; Amazon's computers are drawing inferences from data, not being clairvoyant. But Amazon's guesses are pretty good, and recommending the wrong book every now and then is a very low-cost mistake. If Amazon does it too often, I might switch to Barnes and Noble, but there is no injury to me. So again: Why should anyone care that Amazon knows so much about me? On the surface, it seems benign. Of course, we don't want the credit card information to go astray, but who cares about knowing what books I have looked at online?

Our indifference is another marker of the fact that we are living in an exposed world, and that it feels very different to live here. In 1988, when a

How Sites Know Who You Are

1. **You tell them**. Log in to Gmail, Amazon, or eBay, and you are letting them know exactly who you are.
2. **They've left cookies on one of your previous visits**. A *cookie* is a small text file stored on your local hard drive that contains information that a particular web site wants to have available during your current session (like your shopping cart), or from one session to the next. Cookies give sites persistent information for tracking and personalization. Your browser has a command for showing cookies—you may be surprised how many web sites have left them!
3. **They have your IP address**. The web server has to know where you are so that it can ship its web pages to you. Your IP address is a number like 66.82.9.88 that locates your computer in the Internet (see the Appendix for details). That address may change from one day to the next. But in a residential setting, your Internet Service Provider (your *ISP*—typically your phone or cable company) knows who was assigned each IP address at any time. Those records are often subpoenaed in court cases.

If you are curious about who is using a particular IP address, you can check the American Registry of Internet Numbers (`www.arin.net`). Services such as `whatismyip.com`, `whatismyip.org`, and `ipchicken.com` also allow you to check your own IP address. And `www.whois.net` allows you to check who owns a domain name such as `harvard.com`—which turns out to be the Harvard Bookstore, a privately owned bookstore right across the street from the university. Unfortunately, that information won't reveal who is sending you spam, since spammers routinely forge the source of email they send you.

videotape rental store clerk turned over Robert Bork's movie rental records to a Washington, DC newspaper during Bork's Supreme Court confirmation hearings, Congress was so outraged that it quickly passed a tough privacy protection bill, The Video Privacy Protection Act. Videotape stores, if any still exist, can be fined simply for keeping rental records too long. Twenty years later, few seem to care much what Amazon does with its millions upon millions of detailed, fine-grained views into the brains of all its customers.

It's Just Fun to Be Exposed

Sometimes, there can be no explanation for our willing surrender of our privacy except that we take joy in the very act of exposing ourselves to public

view. Exhibitionism is not a new phenomenon. Its practice today, as in the past, tends to be in the province of the young and the drunk, and those wishing to pretend they are one or the other. That correlation is by no means perfect, however. A university president had to apologize when an image of her threatening a Hispanic male with a stick leaked out from her MySpace page, with a caption indicating that she had to "beat off the Mexicans because they were constantly flirting with my daughter."

And there is a continuum of outrageousness. The less wild of the party photo postings blend seamlessly with the more personal of the blogs, where the bloggers are chatting mostly about their personal feelings. Here there is not exuberance, but some simpler urge for human connectedness. That passion, too, is not new. What *is* new is that a photo or video or diary entry, once posted, is visible to the entire world, and that there is no taking it back. Bits don't fade and they don't yellow. Bits are forever. And we don't know how to live with that.

Bits don't fade and they don't yellow. Bits are forever. And we don't know how to live with that.

For example, a blog selected with no great design begins:

> This is the personal web site of Sarah McAuley. ... I think sharing my life with strangers is odd and narcissistic, which of course is why I'm addicted to it and have been doing it for several years now. Need more? You can read the "About Me" section, drop me an email, or you know, just read the drivel that I pour out on an almost-daily basis.

No thank you, but be our guest. Or consider that there is a Facebook group just for women who want to upload pictures of themselves uncontrollably drunk. Or the Jennicam, through which Jennifer Kay Ringley opened her life to the world for seven years, setting a standard for exposure that many since have surpassed in explicitness, but few have approached in its endless ordinariness. We are still experimenting, both the voyeurs and viewed.

Because You Can't Live Any Other Way

Finally, we give up data about ourselves because we don't have the time, patience, or single-mindedness about privacy that would be required to live our daily lives in another way. In the U.S., the number of credit, debit, and bank cards is in the billions. Every time one is used, an electronic handshake records a few bits of information about who is using it, when, where, and for what. It is now virtually unheard of for people to make large purchases of

ordinary consumer goods with cash. Personal checks are going the way of cassette tape drives, rendered irrelevant by newer technologies. Even if you could pay cash for everything you buy, the tax authorities would have you in their databases anyway. There even have been proposals to put RFIDs in currency notes, so that the movement of cash could be tracked.

Only sects such as the Amish still live without electricity. It will soon be almost that unusual to live without Internet connectivity, with all the fingerprints it leaves of your daily searches and logins and downloads. Even the old dumb TV is rapidly disappearing in favor of digital communications. Digital TV will bring the advantages of video on demand—no more trips to rent movies or waits for them to arrive in the mail—at a price: Your television service provider will record what movies you have ordered. It will be so attractive to be able to watch what we want when we want to watch it, that we won't miss either the inconvenience or the anonymity of the days when all the TV stations washed your house with their airwaves. You couldn't pick the broadcast times, but at least no one knew which waves you were grabbing out of the air.

Little Brother Is Watching

So far, we have discussed losses of privacy due to things for which we could, in principle anyway, blame ourselves. None of us really needs a loyalty card, we should always read the fine print when we rent a car, and so on. We would all be better off saying "no" a little more often to these privacy-busters, but few of us would choose to live the life of constant vigilance that such resolute denial would entail. And even if we were willing to make those sacrifices, there are plenty of other privacy problems caused by things others do to us.

The snoopy neighbor is a classic American stock figure—the busybody who watches how many liquor bottles are in your trash, or tries to figure out whose Mercedes is regularly parked in your driveway, or always seems to know whose children were disorderly last Saturday night. But in Cyberspace, we are all neighbors. We can all check up on each other, without even opening the curtains a crack.

Public Documents Become VERY Public

Some of the snooping is simply what anyone could have done in the past by paying a visit to the Town Hall. Details that were always public—but inaccessible—are quite accessible now.

suspected threats to defense facilities as part of a larger program of domestic counterintelligence.

The Transportation Security Administration (TSA) is responsible for airline passenger screening. One proposed system, CAPPS II, which was ultimately terminated over privacy concerns, sought to bring together disparate data sources to determine whether a particular individual might pose a transportation threat. Color-coded assessment tags would determine whether you could board quickly, be subject to further screening, or denied access to air travel.

The government creates projects, the media and civil liberties groups raise serious privacy concerns, the projects are cancelled, and new ones arise to take their place. The cycle seems to be endless. In spite of Americans' traditional suspicions about government surveillance of their private lives, the cycle seems to be almost an inevitable consequence of Americans' concerns about their security, and the responsibility that government officials feel to use the best available technologies to protect the nation. Corporate databases often contain the best information on the people about whom the government is curious.

Technology Change and Lifestyle Change

New technologies enable new kinds of social interactions. There were no suburban shopping malls before private automobiles became cheap and widely used. Thirty years ago, many people getting off an airplane reached for cigarettes; today, they reach for cell phones. As Heraclitus is reported to have said 2,500 years ago, "all is flux"—everything keeps changing. The reach-for-your-cell phone gesture may not last much longer, since airlines are starting to provide onboard cell phone coverage.

The more people use a new technology, the more useful it becomes. (This is called a "network effect"; see Chapter 4, "Needles in the Haystack.") When one of us got the email address `lewis@harvard` as a second-year graduate student, it was a vainglorious joke; all the people he knew who had email addresses were students in the same office with him. Email culture could not develop until a lot of people had email, but there wasn't much point in having email if no one else did.

Technology changes and social changes reinforce each other. Another way of looking at the technological reasons for our privacy loss is to recognize that the social institutions enabled by the technology are now more important than the practical uses for which the technology was originally conceived. Once a lifestyle change catches on, we don't even think about what it depends on.

Credit Card Culture

The usefulness of the data aggregated by Acxiom and its kindred data aggregation services rises as the number of people in their databases goes up, and as larger parts of their lives leave traces in those databases. When credit cards were mostly short-term loans taken out for large purchases, the credit card data was mostly useful for determining your creditworthiness. It is still useful for that, but now that many people buy virtually everything with credit cards, from new cars to fast-food hamburgers, the credit card transaction database can be mined for a detailed image of our lifestyles. The information is there, for example, to determine if you usually eat dinner out, how much traveling you do, and how much liquor you tend to consume. Credit card companies do in fact analyze this sort of information, and we are glad they do. If you don't seem to have been outside Montana in your entire life and you turn up buying a diamond bracelet in Rio de Janeiro, the credit card company's computer notices the deviation from the norm, and someone may call to be sure it is really you.

The credit card culture is an economic problem for many Americans, who accept more credit card offers than they need, and accumulate more debt than they should. But it is hard to imagine the end of the little plastic cards, unless even smaller RFID tags replace them. Many people carry almost no cash today, and with every easy swipe, a few more bits go into the databases.

Email Culture

Email is culturally in between telephoning and writing a letter. It is quick, like telephoning (and instant messaging is even quicker). It is permanent, like a letter. And like a letter, it waits for the recipient to read it. Email has, to a great extent, replaced both of the other media for person-to-person communication, because it has advantages of both. But it has the problems that other communication methods have, and some new ones of its own.

Phone calls are not intended to last forever, or to be copied and redistributed to dozens of other people, or to turn up in court cases. When we use email as though it were a telephone, we tend to forget about what else might happen to it, other than the telephone-style use, that the recipient will read it and throw it away. Even Bill Gates probably wishes that he had written his corporate emails in a less telephonic voice. After testifying in an antitrust lawsuit that he had not contemplated cutting a deal to divide the web browser market with a competitor, the government produced a candid email he had sent, seeming to contradict his denial: "We could even pay them money as part of the deal, buying a piece of them or something."

Email is as public as postcards, unless it is encrypted, which it usually is not.

Email is bits, traveling within an ISP and through the Internet, using email software that may keep copies, filter it for spam, or submit it to any other form of inspection the ISP may choose. If your email service provider is Google, the point of the inspection is to attach some appropriate advertising. If you are working within a financial services corporation, your emails are probably logged—even the ones to your grandmother—because the company has to be able to go back and do a thorough audit if something inappropriate happens.

Email is as public as postcards, unless it is encrypted, which it usually is not. Employers typically reserve the right to read what is sent through company email. Check the policy of your own employer; it may be hard to find, and it may not say what you expect. Here is Harvard's policy, for example:

> Employees must have no expectation or right of privacy in anything they create, store, send, or receive on Harvard's computers, networks, or telecommunications systems. .... Electronic files, e-mail, data files, images, software, and voice mail may be accessed at any time by management or by other authorized personnel for any business purpose. Access may be requested and arranged through the system(s) user, however, this is not required.

Employers have good reason to retain such sweeping rights; they have to be able to investigate wrongdoing for which the employer would be liable. As a result, such policies are often less important than the good judgment and ethics of those who administer them. Happily, Harvard's are generally good. But as a general principle, the more people who have the authority to snoop, the more likely it is that someone will succumb to the temptation.

Commercial email sites can retain copies of messages even after they have been deleted. And yet, there is very broad acceptance of public, free, email services such as Google's Gmail, Yahoo! Mail, or Microsoft's Hotmail. The technology is readily available to make email private: whether you use encryption tools, or secure email services such as Hushmail, a free, web-based email service that incorporates PGP-based encryption (see Chapter 5). The usage of these services, though, is an insignificant fraction of their unencrypted counterparts. Google gives us free, reliable email service and we, in return, give up some space on our computer screen for ads. Convenience and cost trump privacy. By and large, users don't worry that Google, or its competitors, have all their mail. It's a bit like letting the post office keep a copy of every letter you send, but we are so used to it, we don't even think about it.

Web Culture

When we send an email, we think at least a *little* bit about the impression we are making, because we are sending it to a human being. We may well say things we would not say face-to-face, and live to regret that. Because we can't see anyone's eyes or hear anyone's voice, we are more likely to overreact and be hurtful, angry, or just too smart for our own good. But because email is directed, we don't send email thinking that no one else will ever read what we say.

The Web is different. Its social sites inherit their communication culture not from the letter or telephone call, but from the wall in the public square, littered with broadsides and scribbled notes, some of them signed and some not. Type a comment on a blog, or post a photo on a photo album, and your action can be as anonymous as you wish it to be—you do not know to whom your message is going. YouTube has millions of personal videos. Photo-archiving sites are the shoeboxes and photo albums of the twenty-first century. Online backup now provides easy access to permanent storage for the contents of our personal computers. We entrust commercial entities with much of our most private information, without apparent concern. The generation that has grown up with the Web has embraced social networking in all its varied forms: MySpace, YouTube, LiveJournal, Facebook, Xanga, Classmates.com, Flickr, dozens more, and blogs of every shape and size. More than being taken, personal privacy has been given away quite freely, because everyone else is doing it—the surrender of privacy is more than a way to social connectedness, it is a social institution in its own right. There are 70 million bloggers sharing everything from mindless blather to intimate personal details. Sites like `www.loopt.com` let you find your friends, while `twitter.com` lets you tell the entire world where you are and what you are doing. The Web is a confused, disorganized, chaotic realm, rich in both gold and garbage.

The "old" web, "Web 1.0," as we now refer to it, was just an information resource. You asked to see something, and you got to see it. Part of the disinhibition that happens on the new "Web 2.0" social networking sites is due to the fact that they still allow the movie-screen illusion—that we are "just looking," or if we are contributing, we are not leaving footprints or fingerprints if we use pseudonyms. (See Chapter 4 for more on Web 1.0 and Web 2.0.)

But of course, that is not really the way the Web ever worked. It is important to remember that even Web 1.0 was never anonymous, and even "just looking" leaves fingerprints.

In July 2006, a *New York Times* reporter called Thelma Arnold of Lilburn, Georgia. Thelma wasn't expecting the call. She wasn't famous, nor was she involved in anything particularly noteworthy. She enjoyed her hobbies, helped her friends, and from time to time looked up things on the Web—stuff about her dogs, and her friends' ailments.

Then AOL, the search engine she used, decided to release some "anonymous" query data. Thelma, like most Internet users, may not have known that AOL had kept every single topic that she, and every other one of their users, had asked about. But it did. In a moment of unenlightened generosity, AOL released for research use a small sample: about 20 million queries from 658,000 different users. That is actually not a lot of data by today's standards. For example, in July 2007, there were about 5.6 billion search engine queries, of which roughly 340 million were AOL queries. So, 20 million queries comprise only a couple of days' worth of search queries. In an effort to protect their clients' privacy, AOL "de-identified" the queries. AOL never mentioned anyone by name; they used random numbers instead. Thelma was 4417149. AOL mistakenly presumed that removing a single piece of personal identification would make it hard to figure out who the users were. It turned out that for some of the users, it wasn't hard at all.

It didn't take much effort to match Thelma with her queries. She had searched for "landscapers in Lilburn, GA" and several people with the last name "Arnold," leading to the obvious question of whether there were any Arnolds in Lilburn. Many of Thelma's queries were not particularly useful for identifying her, but were revealing nonetheless: "dry mouth," "thyroid," "dogs that urinate on everything," and "swing sets."

Thelma was not the only person to be identified. User 22690686 (Terri) likes astrology, and the Edison National Bank, Primerica, and Budweiser. 5779844 (Lawanna) was interested in credit reports, and schools. From what he searched for, user 356693 seems to have been an aide to Chris Shays, Congressman from Connecticut.

One of the privacy challenges that we confront as we rummage through the rubble of the digital explosion is that information exists without context. Was Thelma Arnold suffering from a wide range of ailments? One might readily conclude that from her searches. The fact is that she often tried to help her friends by understanding their medical problems.

Or consider AOL user 17556639, whose search history was released along with Thelma Arnold's. He searched for the following:

how to kill your wife 23 Mar, 22:09
wife killer 23 Mar, 22:11
poop 23 Mar, 22:12
dead people 23 Mar, 22:13
pictures of dead people 23 Mar, 22:15
killed people 23 Mar, 22:16
dead pictures 23 Mar, 22:17
murder photo 23 Mar, 22:20
steak and cheese 23 Mar, 22:22
photo of death 23 Mar, 22:30
death 23 Mar, 22:33
dead people photos 23 Mar, 22:33
photo of dead people 23 Mar, 22:35
www.murderdpeople.com 23 Mar, 22:37
decapitated photos 23 Mar, 22:39
car crashes3 23 Mar, 22:40
car crash photo 23 Mar, 22:41

Is this AOL user a potential criminal? Should AOL have called the police? Is 17556639 about to kill his wife? Is he (or she) a researcher with a spelling problem and an interest in Philly cheese steak? Is reporting him to the police doing a public service, or is it an invasion of privacy?

There is no way to tell just from these queries if this user was contemplating some heinous act or doing research for a novel that involves some grisly scenes. When information is incomplete and decontextualized, it is hard to judge meaning and intent.

In this particular case, we happen to know the answer. The user, Jason from New Jersey, was just fooling around, trying to see if Big Brother was watching. He wasn't planning to kill his wife at all. Inference from incomplete data has the problem of false positives—thinking you have something that you don't, because there are other patterns that fit the same data.

Information without context often leads to erroneous conclusions. Because our digital trails are so often retrieved outside the context within which they were created, they sometimes suggest incorrect interpretations. Data interpretation comes with balanced social responsibilities, to protect society when there is evidence of criminal behavior or intent, and also to protect the individual when such evidence is too limited to be reliable. Of course, for every example of misleading and ambiguous data, someone will want to solve the problems it creates by collecting more data, rather than less.

Beyond Privacy

There is nothing new under the sun, and the struggles to define and enforce privacy are no exception. Yet history shows that our concept of privacy has evolved, and the law has evolved with it. With the digital explosion, we have arrived at a moment where further evolution will have to take place rather quickly.

Leave Me Alone

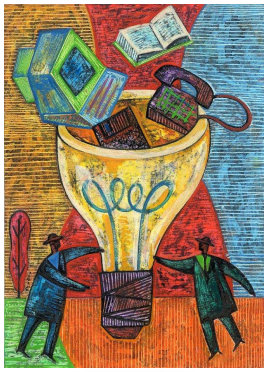
More than a century ago, two lawyers raised the alarm about the impact technology and the media were having on personal privacy:

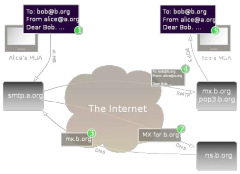
> Instantaneous photographs and newspaper enterprise have invaded the sacred precincts of private and domestic life; and numerous mechanical devices threaten to make good the prediction that "what is whispered in the closet shall be proclaimed from the house-tops."

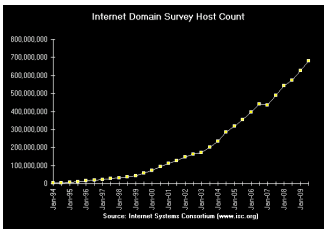
This statement is from the seminal law review article on privacy, published in 1890 by Boston attorney Samuel Warren and his law partner, Louis Brandeis, later to be a justice of the U.S. Supreme Court. Warren and Brandeis went on, "Gossip is no longer the resource of the idle and of the vicious, but has become a trade, which is pursued with industry as well as effrontery. To satisfy a prurient taste the details of sexual relations are spread broadcast in the columns of the daily papers. To occupy the indolent, column upon column is filled with idle gossip, which can only be procured by intrusion upon the domestic circle." New technologies made this garbage easy to produce, and then "the supply creates the demand."

And those candid photographs and gossip columns were not merely tasteless; they were bad. Sounding like modern critics of mindless reality TV, Warren and Brandeis raged that society was going to hell in a handbasket because of all that stuff that was being spread about.

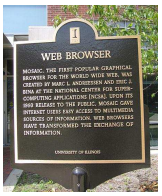
> Even gossip apparently harmless, when widely and persistently circulated, is potent for evil. It both belittles and perverts. It belittles by inverting the relative importance of things, thus dwarfing the thoughts and aspirations of a people. When personal gossip attains the dignity of print, and crowds the space available for matters of

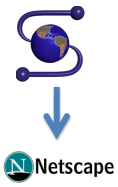


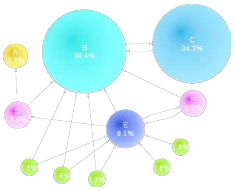


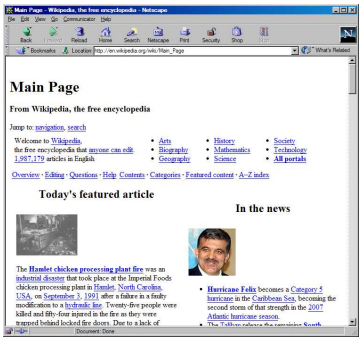


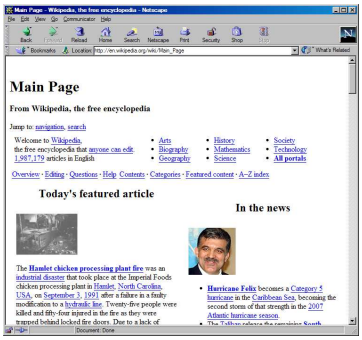
en.wikipedia.org/wiki/Personal_computer

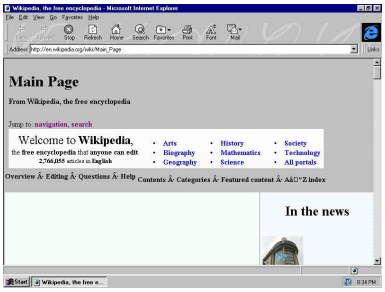

WEB BROWSER
MOSAIC, THE FIRST POPULAR GRAPHICAL BROWSER FOR THE WORLD WIDE WEB, WAS CREATED AT NCSA-UIUC BY MARC L. ANDREESSEN AND ERIC J. BINA. AFTER THE NATIONAL CENTER FOR SUPERCOMPUTING APPLICATIONS (NCSA) UPON ITS 1993 RELEASE TO THE PUBLIC, MOSAIC GAVE INTERNET USERS EASY ACCESS TO MULTIMEDIA SOURCES OF INFORMATION. WEB BROWSERS HAVE TRANSFORMED THE EXCHANGE OF INFORMATION.
UNIVERSITY OF ILLINOIS

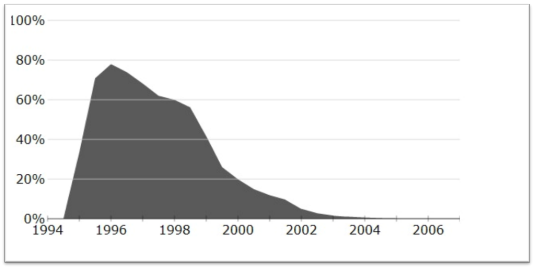

Netscape

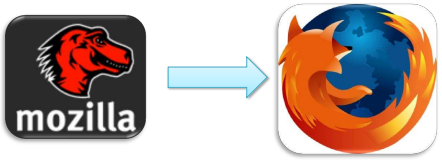


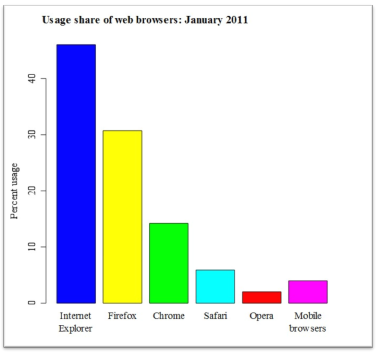

Main Page
From Wikipedia, the free encyclopedia
Today's featured article
In the news


Main Page
From Wikipedia, the free encyclopedia
Today's featured article
In the news

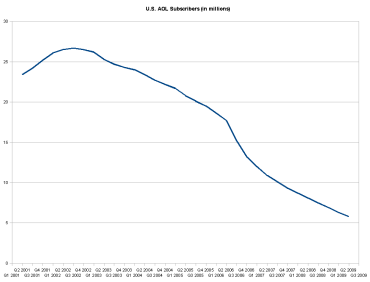

Main Page
From Wikipedia, the free encyclopedia
Welcome to Wikipedia,
In the news


100%
80%
60%
40%
20%
0%
1994
1996
1998
2000
2002
2004
2006


mozilla


Usage share of web browsers: January 2011
Percent usage
0
10
20
30
40
Internet Explorer
Firefox
Chrome
Safari
Opera
Mobile browsers


Channels
Influence
GAMES
TRAVEL
Interests
International
Lifestyles
Personal Finance
Shopping
WorkPlace
Health
AOL Today
Computing
families
NEWS
Research & Learn
KIDSONLY
SPORTS
Entertainment
Local


U.S. AOL Subscribers (in millions)


Web 2.0
Participation
Usability
Folksonomy
Wikis
Blogs
Joy of Use
Recommendation
Social Software
Simplicity
AJAX
Audio
Video
Design
Convergence
CSS
Mobility
Affiliation
OpenAPIs
RSS
Web Standards
Economy
Remixability
Standardization
The Long Tail
DataDriven
Microformats


TIME
You.

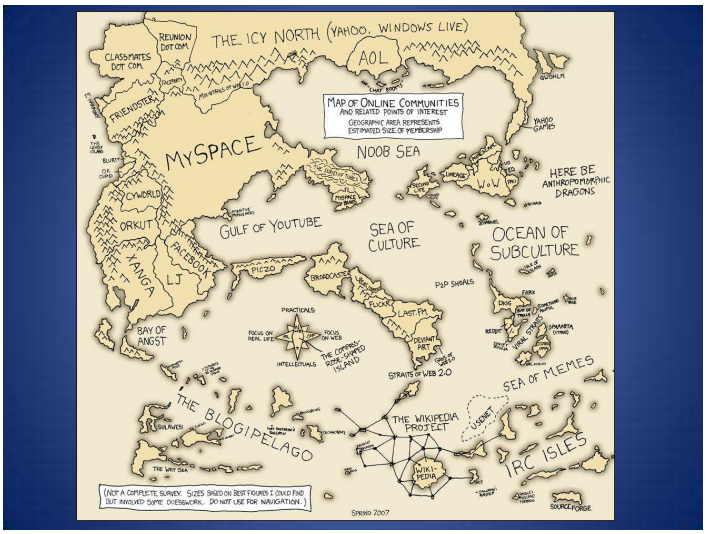
napster

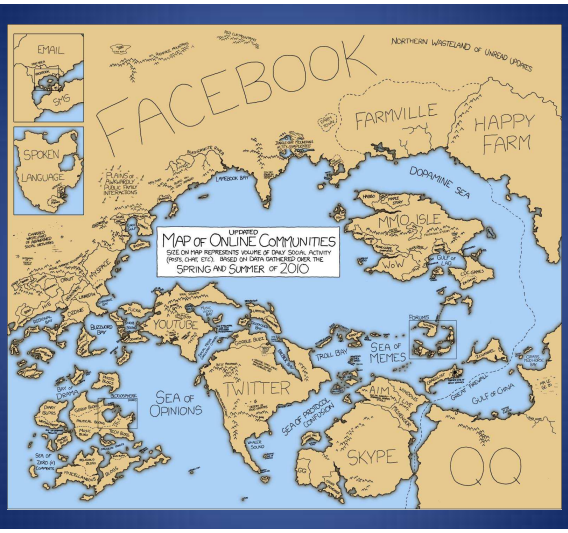
myspace®
a place for friends

friendster

facebook

facebook





Facebook Growth

- http://www.penn-olson.com/2010/02/10/infographic-facebooks-amazing-growth/

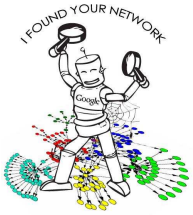
Facebook's Impacts

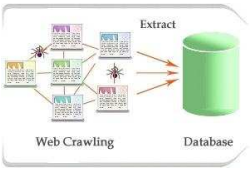
- How has Facebook affected you?
  - Personal Life
  - Academic / Professional Life
- How has Facebook affected the world?
  - Literally, has revolutionized countries and overthrown governments.

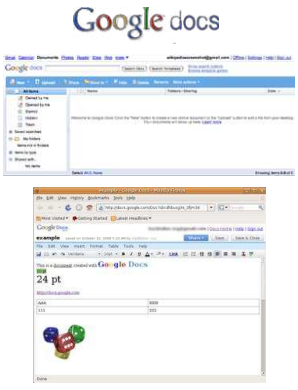
Facebook

- What "shoulders" are they standing on?


Google!
BETA
Search the web using Google!
Google Search
I'm feeling lucky
Special Searches
Stanford Search
Linux Search
Help!
About Google!
Company Info
Google! Logos
Get Google!
updates monthly:
your e-mail
Subscribe
Archive
Copyright ©1998 Google Inc.


I FOUND YOUR NETWORK


Extract
Web Crawling
Database


Google docs
24 pt


Gmail
by Google


31


Google
talk
BETA


Google Docs

http://www.fastcompany.com/1733627/mit-scientist-captures-his-sons-first-90000-hours-on-video

[redacted]

Algorithms

Computer Science Principles

Problem Solving Tools

- Programs are created to solve problems.
- A solution must be designed prior to coding.
- One method of designing a solution to a problem is to create an ***algorithm***.

Algorithms

- An ***algorithm*** is a list of steps to solve a problem written in plain English.
  - **Steps** to solve a problem are written out and numbered in the order in which they should be executed.
- They should be as extensive as necessary to outline the solution.
- Your algorithm is not only going to tell your program what to do but how to do it.

Algorithm Example – Going Home

- The Walk Algorithm
  1. Leave classroom
  2. Turn right out of school building
  3. Walk 1.2 miles
  4. Turn right on street
  5. Go to 4th house
- The Bus Algorithm
  1. Go to the bus area
  2. Get in right bus
  3. Go to house

Algorithms

- Both algorithms, and others that accomplish the same task (of getting you home).
- There are advantages and disadvantages associated with each option.
- You have to consider each option and its advantages/disadvantages before you choose the algorithm you want to continue developing into your program.

Programming Algorithm Example

Simple steps representing a process for dealing with a guessing game in which the computer generates a random number and the player guesses.

1. Generate a secret random number between 1 and 100.
2. Get a number from the player.
3. Compare the player's guess to the secret number.
4. Compare the numbers. If the numbers are identical, go to step 5. Otherwise, tell the player the number was either too high and return to step 2.
5. Display a message stating the secret number was guessed.

Pseudocode

- ***Pseudocode*** is a mix of English language and code that represents what you want your program to do.
- It helps you determine how you want the program to work as well as what variables and methods/functions you will want to include.
- Developing pseudocode will help you work through your logic, reducing the number of errors and potential re-writes you will have to do.

Pseudocode Example

Represents the same process for dealing with a guessing game in which the computer generates a random number and the player guesses the number

```
btnCheckGuess_Click()
   randomNumber = 37
   Get playerGuess from text box
   If playerGuess = randomNumber  Then
      Display "Correct"
   ElseIf playerGuess < randomNumber Then
      Display "Guess too Low"
   Else
      Display "Guess too High"
End
```

Flowchart

- A third tool in programming is through the use of a ***flowchart***.
- Flowcharts use symbols and text to give a visual representation of a solution to a problem.
- The direction of the arrows indicates the flow of the logic.

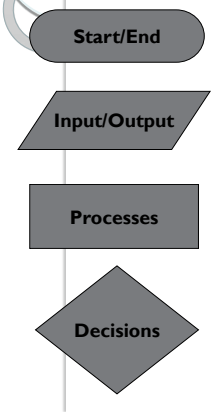
Flowchart

- Flowcharts help the programmer begin to plan the programming project.
- They provide a ***visual representation*** of the algorithm or process.
- They describe the inputs, processes and outputs of the program that are needed to successfully complete the project.

Flowchart Symbols

- There are many flowchart programs, however you can also use Microsoft Word to create a flowchart – or just a piece of paper and a pencil.
- To create the flowchart, there are different symbols that represent the various parts. We will only use a few of these symbols.
- Use lines with arrows to indicate flow of control.
- The text in your flowchart symbols is your pseudocode.

Flowchart Symbols



Ovals : Start should always be the first shape, with a End at the end of the flow chart or a process.

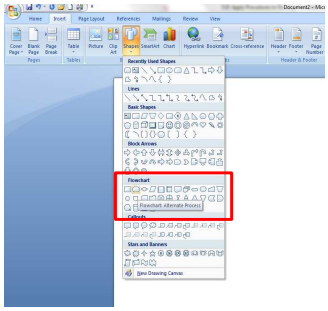
Parallelogram: This shape is used to show raw materials used for 'ingredients' and to show the finished product. Input/Output – Get/Display

Rectangles should be used to show processes/commands, eg. 'Bake Cake'. These are activities.

Diamonds: Hold questions that resolve into True or False. Used for **decisions** that divide into two options and to control **loops**.

Using Microsoft Word to Create a Flowchart

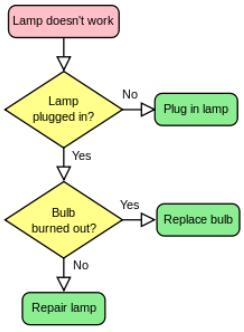
- Open Microsoft Word.
- Under Insert choose Shapes
- Look down the list until you see Flowchart.
- Hover your mouse over a shape, you will see a popup telling you what that shape is used for.



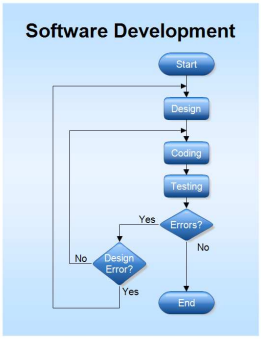
Using Microsoft Word to Create a Flowchart

- Select and draw the shapes needed for your program logic.
- Once you draw a shape you can right click and select Add Text to enter information into your symbol.
- Join your symbols using arrows indicating program data flow.

Flowchart Examples



http://en.wikipedia.org/wiki/Flowchart



http://www.rff.com/software_development.htm

Expectations

- Some general rules:
  - If you do not understand the problem, you probably will not be able to create a solution.
  - Remember to start with the solution in mind.
  - Your program solution should not necessarily look like that of another programmer.
  - Use your tools to help you determine your solution.

Algorithm Development

Computer Science Inside…

The Royal Society of Edinburgh
EPSRC Engineering and Physical Sciences Research Council

What is the similarity between these?

- Cooking recipe
- Downloading software or music
- Car repair manual
- Setting up a music playlist
- Knitting pattern
- Calling a friend on the phone
- Sheet music

UNIVERSITY of GLASGOW

What written instructions have you followed…?

- …to complete a task?
  - Can you give an example?
- Were the instructions easy or difficult to follow?
  - Why? What made them easy/ hard?
    - They made sense?
    - You couldn't understand them?
    - They didn't give you enough information?

UNIVERSITY of GLASGOW

Why discuss lists of instructions here??

- Computer programs are lists of instructions
  - with very particular characteristics
  - known as *algorithms*
- How many of you know of a famous computer error/mistake?
- These are caused by the *wrong instructions* in the program
  - the instructions were interpreted by the computer in a way not intended by the program designer
- We are going to explore how these errors come about

UNIVERSITY of GLASGOW

Following an Algorithm

- Algorithm written on hand-out
  - to draw a picture
- You cannot ask for any help
- Don't look at your classmates work
  - do it by yourself!!!

UNIVERSITY of GLASGOW

Here's the algorithm – follow *exactly!!*

1. Draw a diagonal line
2. Draw another diagonal line connected to the top of the first one
3. Draw a straight line from the point where the diagonal lines meet
4. Draw a horizontal line over the straight line
5. At the bottom of the straight line, draw a curvy line
6. Draw a diagonal line from the bottom of the first diagonal to the straight line
7. Draw a diagonal line from the bottom of the second diagonal to the straight line

UNIVERSITY of GLASGOW

How did the pictures turn out?

- Compare your picture with others' pictures…
  - Were they different?
  - Why?
  - What was difficult about following the instructions
  - What was missing from the instructions?

RESULTS

- Let's look at your results.

Putting all this together…

- This time:
  - write an Algorithm
  - test it yourself
  - get someone else to try it out…
- Can you be sure your algorithm will work ok?

Write & test your algorithm

- The task/problem:
  - make a shape out of paper – one sheet of A4
- Write the *algorithm*
  - Write a set of instructions that explains how to make a paper shape from 1 sheet of A4 paper
- Test it
  - Try out your algorithm – does it work?
  - Note: follow your instructions *as closely as possible*
  - Adjust the instructions if necessary

Following an algorithm

- Hide your shape
- Get into pairs
  - by teaming up with someone ***on the opposite side of the room***
  - move to sit together
  - *Do not* show them your paper shape – hide it!!
- Swap algorithm/instructions with your partner
- Follow your partner's instructions to create their paper shape
- Compare shapes
  - how similar is each 'pair' of shapes?
  - what advice can you give on how to improve the instructions?

What do we know about algorithms?

- What are the key characteristics of a "good" algorithm? Why are they hard to develop?
  - Must be unambiguous
  - Must be correct
  - Must be at the right level of detail
- Also, what did we learn about problems we pick?
  - too large sometimes?

Algorithms are fundamental…

- …to Computer Science, and to society
  - Our electronic devices are teeming with algorithms realised in programming code
  - You perform them *every day, every hour…*
- First algorithms developed by the Greeks
  - e.g. Euclidean algorithm for finding greatest common divisor
- "Algorithm" comes from Al Khwarizmi – Persian astronomer and mathematician

Some activities are not *algorithmic* in nature

- Problem solving
- Human thinking process
- Falling in love
- and so on…
- That is why some might think of these are *hard…!!*
- When we can express the human thinking process as an algorithm, Artificial Intelligence will have truly been created

Conclusions

- Algorithm
  - step-by-step method for accomplishing a task
- Following an algorithm
  - relatively easy
- Finding/designing Algorithms
  - difficult but exciting and fulfilling
  - the designed algorithm contains the intelligence of its developer
- Algorithms are a fundamental part of Computer Programming and of Computing Science

Loops & Variables

Computer Science Principles

VARIABLES

What is a Variable?

- A variable is a named space in memory.
- Think of a mailroom with a large wall of slots for the mail as your memory.
  - This is very simplified of course.
- Each of these slots would be assigned to a variable (by its name) and would hold the values assigned.

Adding Variables

- You can add variables to your program to increase its flexibility.
- The variable allows you to change a value as the script runs.
- To add a variable, select the Variables tab, then click on the Make a Variable button.

Creating a Variable

- When you click on the Make a Variable button, the Variable Name window will open.
- Note you can create the variable for the active sprite only (called a local variable) or for all sprites (called a global variable).

Variable Blocks

- You have multiple variable blocks.
  - Checking the checkbox beside a variable name will display the value of the variable on the stage. (only visible when there is a variable.)
  - set [*variableName*] to ()
    - Sets the value
  - change [*variableName*] by ()
    - Changes the value
  - show variable [*variableName*]
    - Displays the value on the stage.
  - hide variable [*variableName*]
    - Displays the value on the stage.
  - script variables (a)
    - Creates local variables

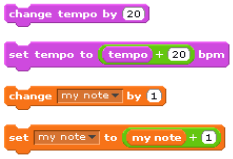
Example of Variable Use

- Create a variable called mynote that will be the value of what note is played.
  - Now the note will change as the loop runs (from using the *repeat ()* block).
  - Note that we had to give the variable a starting value.
    - This is called initializing the variable.

LOOPING

Set vs. Change

- Note that using a set [variableName] to () block will set the value of the variable – NOT update it.
- To update or change a value, use the change [variableName] by () block.

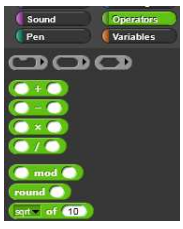


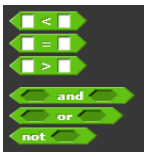
Looping

- There are times when we want certain blocks to repeat more than one time.
- There are blocks that allow us to do just that.
  - *Warp*
    - Does not show the interim steps – only the final product
  - *forever*
    - Will continue to loop until the program closes
    - This is basically an infinite loop as it goes on forever..
  - *repeat ()*
    - Will continue to loop the specified number of times.
  - *repeat until < >*
    - Will continue to loop until the condition is met (*true*)

Helping Blocks

- There are blocks that you will want to use with your variables and loops.
- These blocks are in the Operator's palette.





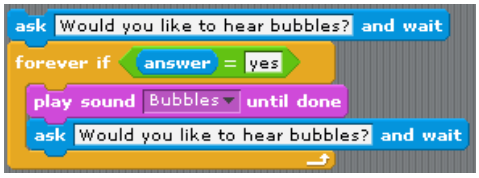
Looping Blocks

- Will continue to play the Bubbles sound.

- Will play the Bubbles sound three times

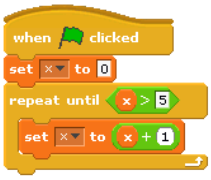
Looping Example

- Will ask the question, then wait for the answer.
- If the answer is "yes" it will play the Bubbles sound.
- Then ask the question again and wait for the answer.
- Playing and asking the question will continue to loop until the answer is something other than "yes"



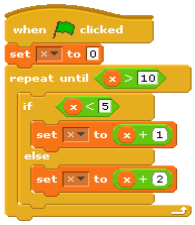
Looping Example

- Let's look at the "repeat until" block a bit closer.
- Just like REPEAT, it will do everything inside the C-shaped block a certain number of times.
- However before it starts the loop each time, it checks to see if the condition (x > 5) is true.
- When this is condition is true, it will not repeat again.



	x
Before the loop	0
Top of loop	0
Bottom of loop	1
Top of loop	1
Bottom of loop	2
Top of loop	2
Bottom of loop	3
Top of loop	3
Bottom of loop	4
Top of loop	4
Bottom of loop	5
Top of loop	5
Bottom of loop	6

repeat until <>

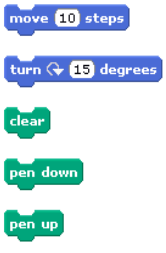


Value of x

DRAWING BLOCKS

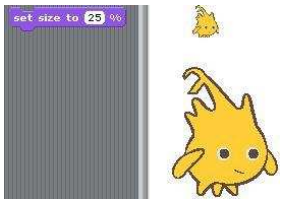
Important Blocks for Drawing

- There are several blocks you will use to draw.
  - *move () steps*
    - Will move your sprite which will draw for you.
  - *turn () degrees*
    - Will turn your sprite to face that direction
  - *clear*
    - Will clear your stage
  - *pen down*
    - Will tell the sprite to start drawing
  - *pen up*
    - Will tell the sprite to stop drawing



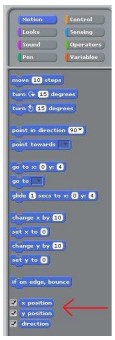
Changing Sprite Size

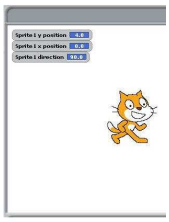
- In order to see your drawing, you might want to change the size of your sprite.
- In the Looks area, you will set the *set size to () %* block.



Where is my Sprite?

- You might also need to know where your sprite is located by the x and y positions as well as the direction your sprite is facing on the stage.
- Look in the Motion area, you will see the several blocks you can use.
- By checking these blocks, the information will be displayed on the stage.



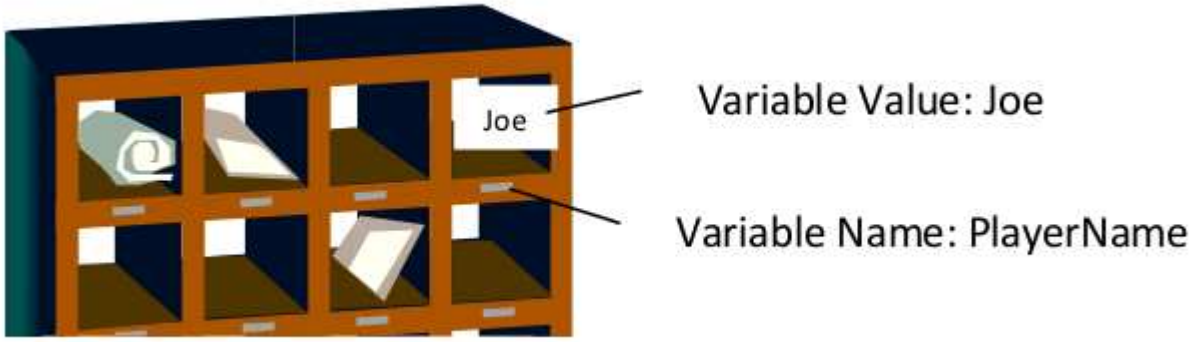


Lab: Variables and looping

Let's open up SNAP at http://snap.berkeley.edu/run (http://snap.berkeley.edu/run)

Variables

Variables are named spaces in memory that your program can access. You can set and modify the values that are contained in these named spaces. To visualize a variable's name space think of mail slots in a large mail room. Your computer has memory that the program is going to use to create and store information. When you create a variable, you are assigning one of the "slots" of memory to a name and then can put a value in that slot and modify it as needed.



Why create variables? Variables allow the programmer to make the value modifiable in the script. For example, you want to be able to update a score variable as the player wins/loses in a game. You will see many different uses of variables during this course.

You have multiple blocks to create and manipulate variables in the Variables tab palette.

- `Make a variable` button allows you to create a new variable
- `Delete a variable` button will allow you to delete a button – this button only shows after you have created a variable.
- `Set [] to (0)` will allow you to initialize, or set a beginning value, the variable to a value.
- `Change [] by (1)` allows you to modify the value of a variable
- `Show variable []` will show the variable and value on the stage.
- `hide variable []` will hide the variable and value on the stage.
- `Script variables (a)` will allow you to create local variables, more on this use later

To create a variable

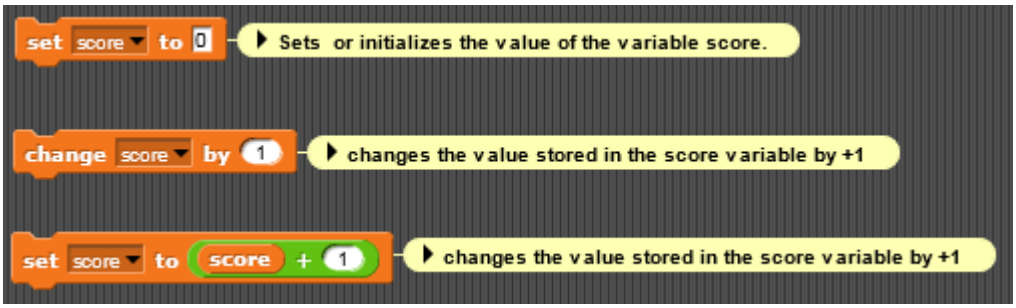
- In the box that pops up - type the name of the variable. The default selection "for all sprites" means that all sprites have access to this variable. Select "for this sprite only" if you want only the sprite currently selected to be able to access/modify the value of the variable.

- Now you'll have blocks to use for your variable.
- Note that you now have a rounded button with the name of the variable in the window with a checkbox to the left. If checked, the variable and value will be show on the stage. Uncheck to hide.
- When you use one of the variable blocks you will be able to click on the combo box arrow for a list of your variables.

Common Bug: Set vs. Change

A very common mistake is to use a change block when you need a set block or use a set block when you need a change block.

It is actually a little more complicated, in that you can make your set block act like a change block.

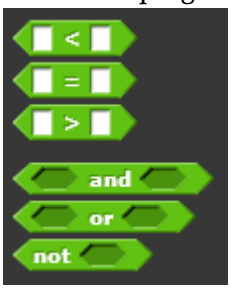


Looping Blocks

There are times when you will want blocks to repeat. Instead of duplicating blocks and ending up with a long script that might be confusing, there are looping control blocks that you can wrap around the script you want to repeat.

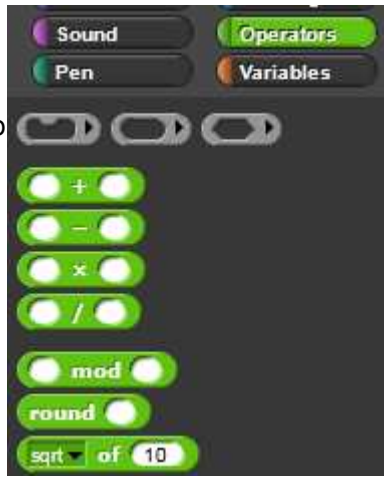
- `forever` Loops until the program ends. This is basically an *infinite loop* as it goes on forever.
- `repeat ()` Loops the specified number of times.
- `repeat until &lt; &gt;` Repeat until the condition is *True*.

For the `repeat until &lt; &gt;` you will use a predicate block that returns true or false. These blocks have pointed ends and can be found in the Operators palette.



Other helpful blocks include the operator blocks.

Operators



Click the Operators tab to display a new palette of blocks. You can use these blocks to perform mathematic operations to modify a numeric variable.

You have blocks to add, subtract, multiple and divide. You also have a mod block that does remainder division as well as a round block and a square root block.

NOTE: To see what any block does, right click on a block in the palette and select help. A new window will open that will explain what that block does.

Looping Examples

Plays the sound continuously until the program is stopped.

Plays the sound completely 10 times then stops

Plays the sound until the `score` variable's value is equal to 10

Repeat Until

Let's look at the "repeat until" block a bit closer. Just like REPEAT, it will do everything inside the C- shaped block a certain number of times. However before it starts the loop each time, it checks to see if the condition (x > 5) is true. When this is condition is true, it will not repeat again.

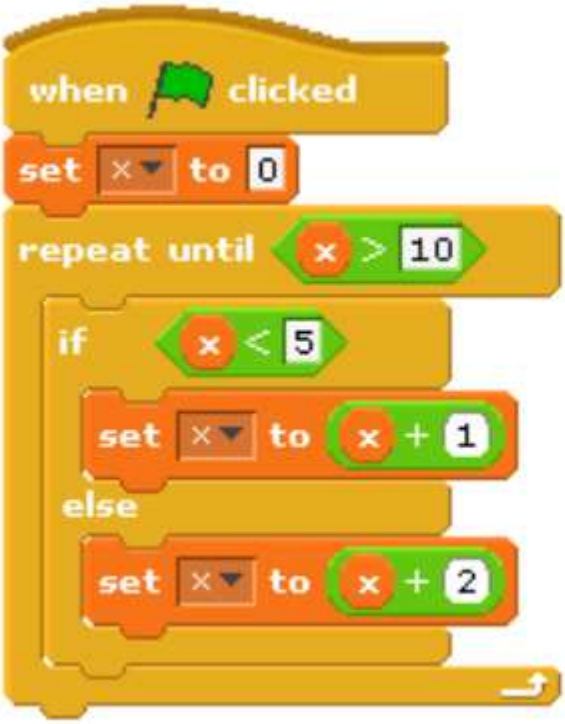
It can be really helpful to keep track of what the variable X is at each point to help us understand how this new piece

works.

- In the right column we keep track of the value of x.
- In the diagram below we draw a horizontal line every time we start the loop. Here we labeled each line “Top of loop” and “Bottom of loop” but we could just use the horizontal line to keep track of this information.
- Within each loop the variable x increases by 1, so we write down the new value for x.

	x
Before the loop	0
Top of loop	0
Bottom of loop	1
Top of loop	1
Bottom of loop	2
Top of loop	2
Bottom of loop	3
Top of loop	3
Bottom of loop	4
Top of loop	4
Bottom of loop	5
Top of loop	5
Bottom of loop	6

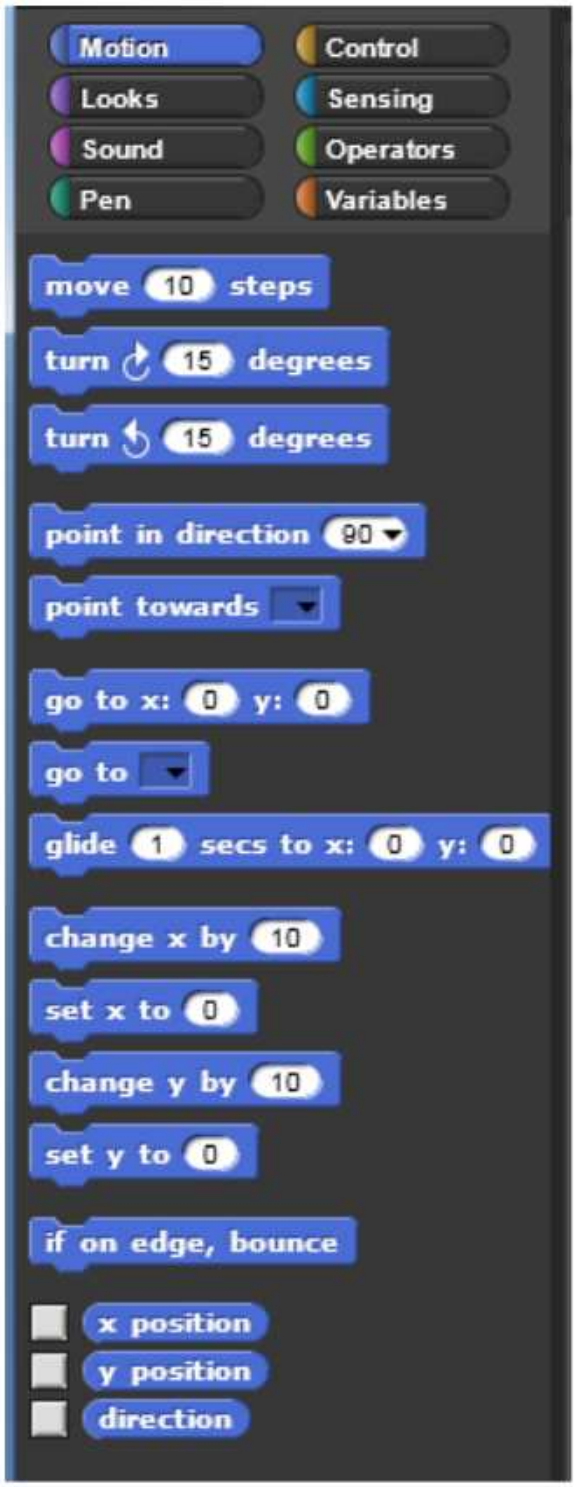
Try to use a chart like the one above to keep track of what happens in the complicated “repeat until” code below.



Value of x

Drawing Tools

There are multiple blocks that can be used to draw. By combining blocks from the Pen palette with blocks from the Motion palette, you can draw pictures. Your sprite needs to face in the direction you want the line to be drawn so you will need the point in direction () block.



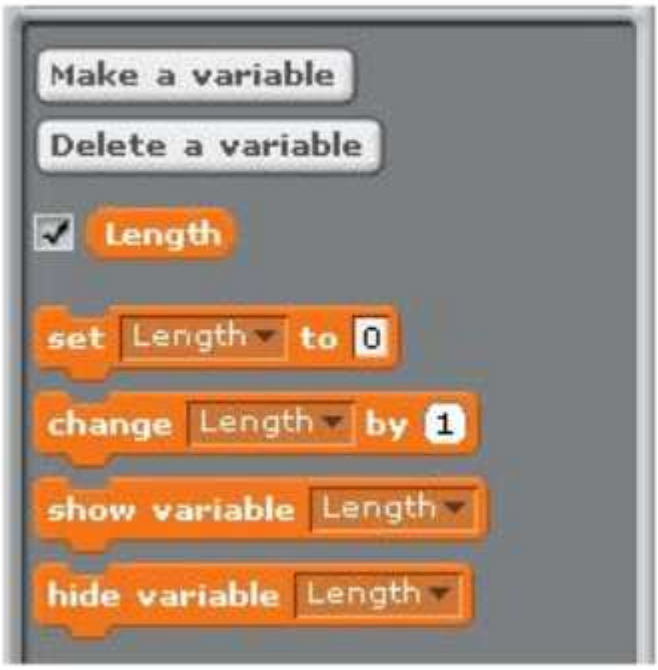
Try it! Drawing shapes

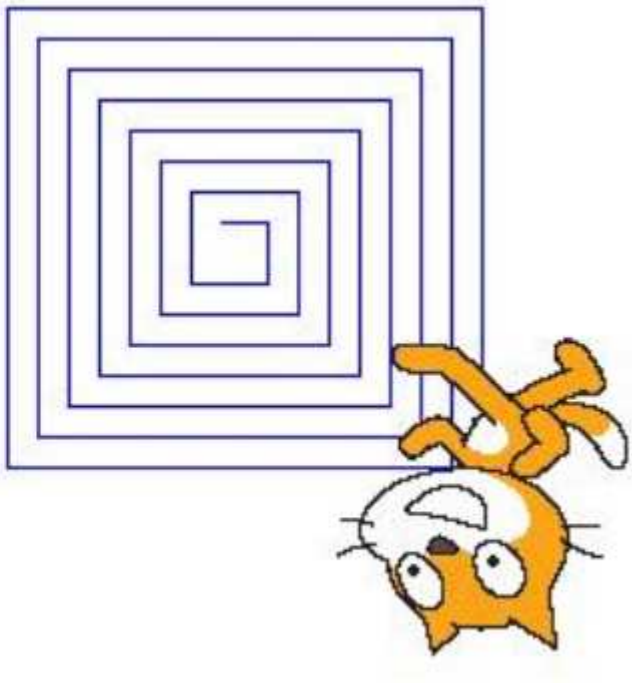
Look at the following scripts to draw a square. The first script has repetitive code. In the second script, the repeating code has been replaced by using a loop.

Note: By using the `pen down` block your sprite will draw for you.

Exercise: Draw a Squaril

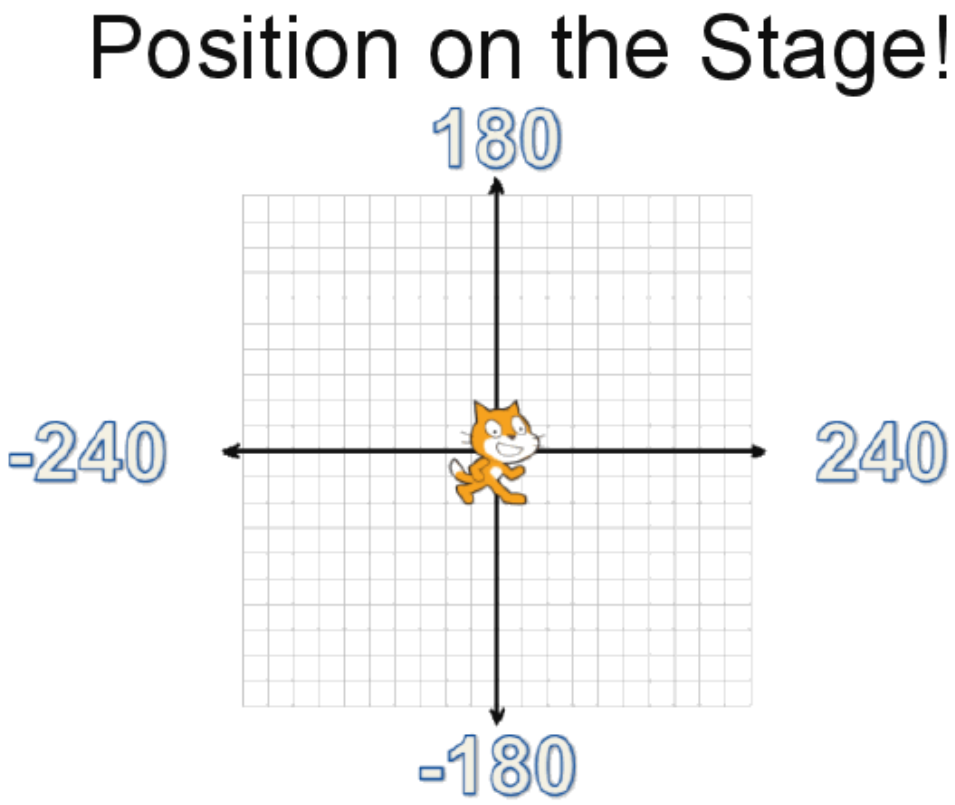
Make a new variable Length with a looping structure to draw the “Squiral” (Square + Spiral) below. Note that the length the sprite moves is updating by a constant amount.





Exercise: Position on the stage

The sprite occupies a point(x,y) on the stage corresponding to the x- and y- axis. Here's a picture:



In Scratch, the sprite usually begins at the center of the stage. Its position is **(0,0)**.

The next lesson asks you to move a random character around the stage within certain boundaries. Keep in mind the coordinate system shown above.

Computer Science Principles Name: _____
Practice Repeat Until Partner Name: _____

Use the script block to evaluate the value of x.

Example

Iterations	x
1	1
2	2
3	3
4	4
5	5
6	7
7	9
8	11

Script in Pseudocode

```
When Flag Clicked
  Set x to 0
  Repeat until x > 10
     if x < 5
        set x to x + 1
    else
        set x to x + 2
  End Loop
```

What is the value of x after the repeat block is finished: ___11____

1. Try this

Iterations	x

b. How many times did this iterate: _____

c. How many times did you enter the “if' block: _____

2. Try this

Iterations	x

b. How many times did this iterate: _____

3. Not so Crazy

a) Make up a not too crazy repeat until block.

b) Switch with a classmate and have them evaluate your script.

c) Draw the script here: (You can use Pseudocode)

d) Evaluate it here:

Iterations	x

4. Evil repeat-until

---

a) Make up an evil repeat until block.

b) Switch with a classmate and have them evaluate your script.

c) Draw the script here: (You can use Pseudocode)

d) Evaluate it here:

Iterations	x

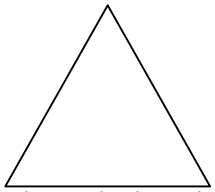
e) How many times did this iterate: _____

Name: _____
Beauty and Joy of Computing: Drawing Regular Figures

In Geometry, you learned (or will learn) how to calculate the exterior angle of any regular-polygon. In this lesson you will draw a series of regular figures. We want to take a second to make sure understand the math behind this.

We'll take it one figure at a time:

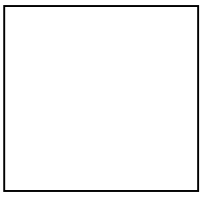
Step 1. The Triangle:



a. Sum of the interior angles in a triangle: 180º
b. Measure of each interior angle : _____
c. Measure of each exterior angle: _____ <== This is the key!
d. Sum of the exterior angles : _____

Step 2: Scratch It …. Now draw the triangle.

Step 3: The Square:



a. Measure of each interior angle: _____
b. Measure of each exterior angle: _____
c. Sum of the exterior angles: _____

Step 4: Scratch It … Now draw the sqaure

Step 5: Putting it together:

a. What do you notice about the sum of the exterior angles in the triangle and square:

b. How is the sum of the exterior angles, the number of sides, and the measure of each exterior angle related:

Step 6: Extend to any figure …

. Given what you figured out in Step 5, finish the chart

# sides	Sum of Exterior Angles	Measure of exterior angle
3		
4		
5		
6		
7		
8		
9		
10		
1000		
n		

Step 7: Scratch It … Draw the figures above

Step 8: The five-sided star is tougher … think about what regular-polygon would circumscribe the five-pointed star.